Michael Ng (SBN 237915)
Michael.Ng@kobrekim.com
Daniel Zaheer (SBN 237118)
Daniel.Zaheer@kobrekim.com
KOBRE & KIM LLP
150 California Street, 19th Floor
San Francisco, CA 94111
Telephone: (415) 582-4800
Fax: (415) 582-4811

Attorneys for Plaintiff
SANAS.AI INC.

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SANAS.AI INC.,
a Delaware corporation,

Plaintiff,

v.

KRISP TECHNOLOGIES, INC.,
a Delaware corporation,

Defendant.

Case No. _____

COMPLAINT FOR DECLARATORY JUDGMENT, PATENT INFRINGEMENT, TRADE SECRET MISAPPROPRIATION, AND FALSE ADVERTISING

DEMAND FOR JURY TRIAL

Plaintiff Sanas.AI Inc., a Delaware corporation ("Plaintiff" or "Sanas"), by and through their attorneys, bring this Complaint against defendant Krisp Technologies, Inc. ("Defendant" or "Krisp") and allege as follows:

INTRODUCTION

Sanas' Revolutionary Accent Translation Technology

1. Sanas is dedicated to helping humanity communicate better, by developing technologies that allow the people of the world to understand and be understood when speaking to each other.

2. The company's inventions have been revolutionary. Sanas uses artificial intelligence tools to break down barriers that prevent even native, fluent speakers from effectively communicating. The company's accent translation technology was the first that translates accents in real time, converting regional accents into local ones, thereby minimizing difficulties that hinder conversation between speakers of the same language from different countries or regions.

3. The company's technical innovation has received widespread acclaim for its smooth, real-time operation, low latency, and for preserving the individual speaker's authenticity and natural intonation. Among other prizes, Sanas received Frost & Sullivan's North American Technology Innovation Leadership Award in 2024, for its "real-time Accent Translation solution built on patented, AI-powered technology that enables users to control how they sound," and which "[u]nlike competing offerings . . . maintains the authenticity of the agent's voice, ensuring a more natural and genuine communication experience."[1]

4. Sanas' innovations have also been lauded for their positive social impact. Sanas' tools have helped overcome discrimination based on accent, and have paved the way for thousands of highly qualified professionals who might otherwise be held back based on the fact that they speak a common language in a different way. In a recent Washington Post article focused on

[1] www.frost.com/news/press-releases/frost-sullivan-recognizes-sanas-as-the-2024-north-american-technology-innovation-leader/

Sanas' innovations, one industry insider noted that such AI technologies are not eliminating the need for people: "What we are seeing is that it is adding value to individual people."[2]

5. Sanas has won rapid adoption in the market and praise from its customers. As the same Washington Post article states about Sanas' accent translation system: "[C]ompanies say it's delivering results: happier customers, satisfied agents, faster calls."[3] Sanas has been used by tens of thousands of customers to facilitate spoken communications and has rapidly become the recognized market leader in the new product category that Sanas created.

Krisp's Copycat Technology

6. Unfortunately, such innovation invites copycats. And that is exactly what happened here.

7. On March 25, 2025, Defendant Krisp launched what it called "Krisp AI Accent Conversion v3," a product it claims is built on its "unique" approach to accent conversion, including real-time operation and preservation of speaker identity.[4]

8. Krisp's product bears a striking resemblance to Sanas' invention. The structure of its system, and the methodologies utilized to implement accent conversion, mimic those that Sanas invented. Even the words Krisp uses to describe what its product is and how it works are ripped from Sanas' marketing and technical materials.

9. The similarity is not random chance. Krisp did not come up with what it claims is its own, "new" accent translation software—Krisp stole it from Sanas.

THE PARTIES

10. Plaintiff Sanas.AI Inc. is a corporation organized under the laws of the State of Delaware, with its principal place of business in this district at 437 Lytton Ave Ste 200, Palo Alto, California, 94301.

[2] www.washingtonpost.com/world/2025/06/21/india-ai-bpo-call-centers/
[3] *Id.*
[4] https://krisp.ai/blog/krisp-ai-accent-conversion-v3/

11. Upon information and belief, Defendant Krisp Technologies, Inc. is a corporation organized under the laws of the State of Delaware, with its principal place of business in this district at 2150 Shattuck Ave. Suite 1300, Berkeley, California, 94704.

JURISDICTION AND VENUE

12. This action arises under the patent laws of the United States, 35 U.S.C. § 271 *et seq.* The Court has subject matter jurisdiction pursuant to 28 U.S.C. §§ 1331, 1338(a), 2201, and/or 2202, based on the actual controversy between Sanas and Krisp arising under the patent laws of the United States, 35 U.S.C. § 100 *et seq*.

13. This Court has supplemental jurisdiction over the declaratory judgment and state law claims herein pursuant to 28 U.S.C. § 1367.

14. This Court has personal jurisdiction over Krisp and venue is proper in this judicial district pursuant to 28 U.S.C. §§1391(b), (c), (d) and 1400(b) because Krisp has a permanent and continuous presence in, has committed acts of infringement in, and maintains a regular and established place of business in this district.

FACTUAL BACKGROUND

Sanas' Origins

15. Like many leading technology companies, Sanas originated with a conversation between friends, in a dorm room, about a real-world problem they faced.

16. In 2020, a Stanford student returned for a stint home in Nicaragua. To help his family there, he took a temporary job answering customer service calls for an international company. He spoke perfect English—he was thriving as a college student at a top American university. But even he was held back in the workplace by his accent, which made effective communication cumbersome.

17. Upon his return to Stanford, he relayed his frustrations to his fellow students. His experience intrigued three of his friends, who wondered if a solution might lie in the new technologies they were studying. Could artificial intelligence be used to help humans communicate more clearly with each other?

18. Together, Maxim Serebryakov, Shawn Zhang, and Andrés Pérez Soderi set about looking for a solution, applying new techniques to the problem and generating the first ideas that would eventually become Sanas.

19. Their goal was straightforward: to use artificial intelligence to allow speakers with one regional accent to be understood by listeners accustomed to a different accent, without sacrificing the individual intonation, cadence, and other nuances that convey meaning with both subtlety and depth in spoken language.

20. Their initial efforts proved promising, so they plunged in, dedicating themselves to creating the first accent translation technology.

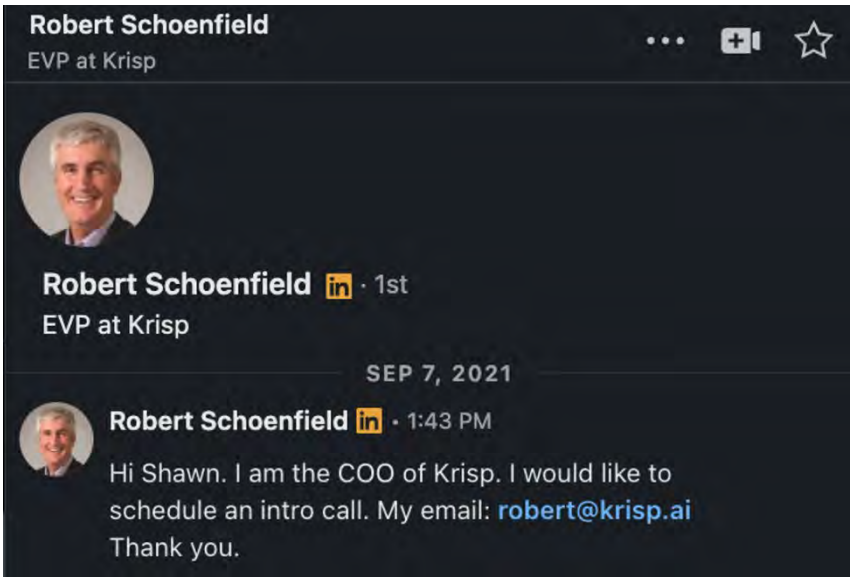
21. As the trio built out the technology, they were joined by Sharath Keshava Narayana, an acclaimed entrepreneur who joined them as a co-founder. In 2021, Sanas launched its first product, and quickly found eager investors in some of Silicon Valley's leading venture capital firms. In 2022, they scaled up, raising a $32 million Series A investment round, including investment from Google.

22. By 2023, Sanas was being used by more than 5,000 contact center agents across 25 cities. Over the next two years, as it expanded its operations, the company found even more success in the market. As of today, Sanas' software has been used by hundreds of thousands of agents.

23. From the outset, Sanas has understood the need to protect its inventions, and has diligently sought patent protection for them. On May 6, 2021, Sanas filed Provisional Patent Application #63/185,345 (Real-Time Accent Conversion Model). This was followed by numerous other applications which disclosed and protected the innovations that solved major problems in execution and implementation of Sanas' market-leading accent translation product.

Krisp Seeks Sanas' Accent Translation Technology

24. On September 7, 2021, Sanas co-founder Shawn Zhang received the following LinkedIn message from Krisp's COO, Robert Schoenfield:



25. The message was unexpected, unsolicited, and unannounced. Sanas and Krisp had no prior direct contact, though Sanas was aware of Krisp.

26. Originally founded as 2Hz and with operations based in Armenia, Krisp had been focused on noise cancellation. According to a blog post from Krisp CEO and co-founder Davit Baghdasaryan in October 2018, the company's objective was "a technology which will completely mute the background noise in human-to-human communications, making it more pleasant and intelligible."[5]

27. Without receiving a response, Schoenfield followed up again with Zhang the next day:

28. Sanas was curious about the outreach. Krisp had made some inroads with the call center market, selling its product as a way to reduce background noise in those sometimes

[5] https://developer.nvidia.com/blog/nvidia-real-time-noise-suppression-deep-learning/

crowded working environments. Sanas thought there was some possibility that the two companies might be complimentary.

29. Krisp continued to press to connect with Sanas. On September 27, 2021, Krisp CEO Baghdasaryan reached out to one of Sanas' venture capital investors, congratulating him on his firm's investment in Sanas and asking if he could help set up a meeting, saying: "We are quite interested in Accent Reduction technologies . . . I would love to connect with the founders and start exploring future partnerships."

30. Through that contact, on October 6, 2021, Baghdasaryan held an introductory Zoom meeting with Sanas' Andrés Pérez Soderi and Maxim Serebryakov. Baghdasaryan followed up enthusiastically, reaching out to Pérez and Serebryakov seeking to meet face-to-face four days later. The Sanas team could not make the last-minute scheduling work, but Baghdasaryan sought another Zoom meeting, saying "this time a deeper one. The main question we have is how this works in real time. Experiencing it somehow would be really important to us."

31. Sanas told him they were willing to conduct a product demonstration and asked for the two parties to enter into a non-disclosure agreement. Krisp agreed, and Krisp's COO Schoenfield sent an NDA, which the parties executed on November 17, 2021.

Krisp and Sanas' Year-Long Partnership Discussions and Technical Evaluation

32. Over the course of the next year, and under the umbrella of their NDA, the two companies explored a possible collaboration. From Sanas' perspective, Krisp had developed a complimentary technology that could supplement its offering to the market, and Krisp had begun rolling out to customers, developing a market presence that could help accelerate Sanas' sales efforts.

33. And from the outset, that collaborative partnership is expressly how Krisp represented its interest as well. COO Schoenfield's first cold outreach stated that Krisp had "100's of global contact centers as customers for background noise removal and voice quality enhancements," and said he was interested in talking to Sanas because "I would like to explore licensing your tech and share more about our business and customers." When CEO Baghdasaryan

made his initial overture through Sanas' investor, he said "We are quite interested in Accent Reduction technologies . . . I would love to connect with the founders and *start exploring future partnerships*." (emphasis added).

34. Through both its words and actions, Krisp told Sanas that it sought out a collaborative partnership through which it would license Sanas' accent translation technology, and to explore ways the two companies might improve the marketing of their respective accent translation (Sanas) and noise reduction (Krisp) offerings.

35. The NDA signed by the parties reflected that. Krisp agreed (as did Sanas) that information it learned in the discussions could not be used without permission for any purpose other than the potential collaboration.

36. One thing was clear: at the time, ***Krisp did not have any accent translation capabilities of its own***.

37. After the parties signed the NDA, Sanas demonstrated its accent translation product and began discussions between the senior business leadership of the companies to explore the possibility of a collaborative approach. A December 1, 2021 meeting included co-founders from both Sanas and Krisp: Andrés Pérez Soderi and Maxim Serebryakov from Sanas along with Krisp co-founders Davit Baghdasaryan and Artavazd Minasyan, as well as Krisp's COO, Robert Schoenfield, and Sanas' Head of Operations Ashley Walker.

38. As those discussions progressed, Baghdasaryan, as Krisp's CEO, reported back that the company wanted to move ahead with the collaboration. On January 13, 2022, Baghdasaryan wrote: "We discussed this internally and think there is a great opportunity to work together. I would like to take this a step further and introduce our engineering teams."

39. At the time, Sanas' leadership team was busy completing its Series A fundraising round, but Krisp kept pushing. On April 13, Krisp's Schoenfield reached back out, resulting in direct discussions with Sanas' co-founder and then-COO, Sharath Keshava Narayana. After further discussions, Schoenfield sent Keshava a proposed letter of intent (LOI) on June 8, 2022, laying out the parameters for the proposed collaboration.

40. On June 13, 2022, Krisp CEO Baghdasaryan sent Sanas' Serebryakov an email laying out the parameters of a "technical evaluation" he wanted to do and "perform . . . a fast-track evaluation" that same week. Specifically, he wanted the technical evaluation to focus on the minutia of performance, compatibility, and implementation:

> "There are several things we would like to test:
> - objective evaluation of accent tech
> - how robust is the tech for diff acoustic conditions (headsets, echo, noise, etc)
> - how compatible are Krisp's noise cancellation and Sanas tech
> - min h/w requirements to run the tech (cpu, memory, disc)"

41. The Krisp team also proposed creating a Slack channel—to be controlled by Krisp—as the forum for the technical exchange of information. Operating under the NDA, Sanas agreed.

42. The senior leadership continued discussions about the terms of a commercial partnership, including "integration approaches" and economic terms. By late summer, Krisp told Sanas that it wanted to move ahead with the partnership.

43. In an email dated August 16, 2022, Krisp's Schoenfield told Sanas' Keshava and Serebryakov:

> "Sharath and Max,
> We spent time this past week with our teams looking at our technical and commercial approach for accent translation. The short of it is that we want to proceed with our partnership."

44. But as they held out prospects for a collaborative partnership, Krisp continued pushing Sanas to provide it with increasingly detailed technical information, representing that it was necessary to assess performance in order to bring the Sanas product to their existing customers. For example, in the same email, Schoenfield laid out Krisp's demand for specific details across multiple conditions:

> "Here is a partial list of our open questions regarding the technical evaluation:
> - performance in case of strong accent, no accent, different dialects
> - intelligibility of the converted audio (converted voice quality, robotic or not)

- performance in case of noise conditions SNR -5db
- what happens if there is another background voice
- quality in case of different speaking pace - wpm
- how it works if we apply Krisp NC, VC before/after Sanas technology?
- end to end latency with different call center platforms
- support of different microphones and cases in reverberated audio
- CPU utilization"

45. Sanas was open to the potential benefits of the prospective relationship but appropriately wary of a full disclosure without a commitment. Sanas CEO Keshava explained that Sanas would have liked to have included an exclusivity provision in the letter of intent to "make[] it easier for [the parties'] technology teams to work together" but explained that he was satisfied that the parties' NDA would provide sufficient protection for the technical discussions.

46. Mr. Keshava also stated that, "I wanted to reiterate the primary reason for both of us to partner should be to assert market dominance . . . As long as we can commit to each other on volumes we would chase for next year by end of Dec I am fine with the construct of the partnership." Schoenfield explained that "the primary reason" to pursue the partnership was "to get to market and secure a first-mover footprint in the industry."

47. Over the ensuing weeks, Krisp continued to push aggressively for the technical teams of the two companies to delve into the details and exchange engineering information directly. On August 21, 2022, Krisp told Sanas that company co-founder Artavazd Minasyan would "lead our technical assessment project" and also brought Stepan Sargsyan, the company's chief scientist, into the discussions.

48. The technical teams from both companies then set about working closely to comb through the details of the Sanas system. On August 23, 2022, for example, Krisp's engineering team sent another list of detailed technical questions they wanted from Sanas:

- "Can a real-time demo of the technology be scheduled?
- Is there a web API we could call or can we somehow test your SDK?
- What is the algorithmic look ahead of your technology? Any estimates for end-to-end latency with different call center platforms?
- What are the estimates for Flops / CPU utilization?
- What is the performance in case of noisy input (like -5 or 0 dB SNR) and in case of reverberated speech?
- How is the technology performing with different microphones? Some

microphones are doing internal signal processing which can impact your algorithm performance.

- What happens if there is another background voice(s)?
- Are the input speaker voice characteristics retained? If not, then how is the generated voice based on the input voice chosen?
- Are there any requirements for input audio bandwidth (narrowband, wideband or full band) and what is the bandwidth of converted speech?
- Do you evaluate the voice quality, acoustic quality, intelligibility, and accentedness of output speech? Are there any objective or subjective evaluation scores of the technology?
- Which accents are supported today?
- What is the performance in case of strong accent and no accent cases?
- Is the technology robust to accented speech variations due to various Indian dialects?
- Is the technology robust to different speaking rates -wpm?
- Are input speaker emotions, prosody or laughter in converted speech retained?"

49. Although Sanas had reservations about the breadth and depth of these inquiries, it provided answers across numerous calls and meetings and in the dedicated Slack channel.

50. The information provided by Sanas to Krisp included information that was at the time nonpublic and highly valuable, as it reflected Sanas' years of development of a new and unprecedented technology; Sanas' solutions to key technological challenges in developing a commercially viable accent translation solution; and information validating that doing so was both achievable and had been achieved through Sanas' innovations.

51. The information disclosed included but is not limited to information concerning:

- Real world performance indicators of Sanas' developed accent translation software, including for example:
  - CPU utilization;
  - End-to-end latency on call-center platforms;
  - Performance in noisy environments;
  - Performance of the software using different types of microphones;
  - Performance of the software in strong and no accent cases;
  - Performance with high speaking rates; and
  - Robustness to laughter;

- Challenges with accent translation software that Sanas had identified as areas of focus for the deployment of development resources, including for example relating to:
  - Performance in noisy environments;
  - Performance problems created by certain types of input sounds;

- Whether Sanas preprocesses audio;
- Approaches for evaluating voice quality, acoustic quality, intelligibility, and accentedness of speech;
- Specific algorithm design tradeoffs;
- Sanas' development of a teacher-student architecture;
- That Sanas' product runs locally on a personal computer CPU; and
- Sanas' use of a parallel speech data processing model.

52. Sanas also provided a demo of its product in action to Krisp.

53. For its part, Krisp expressed gratitude for Sanas' participation in the discussions, while continuing to press for additional disclosures.

54. By the fall, the teams had moved on to the details of integrating the two companies' systems, and discussions about piloting at a select group of customers. Business discussions also appeared to be progressing well, with both remote and in-person meetings between the senior leadership.

Krisp Terminates the Discussions and Copies Sanas' Product

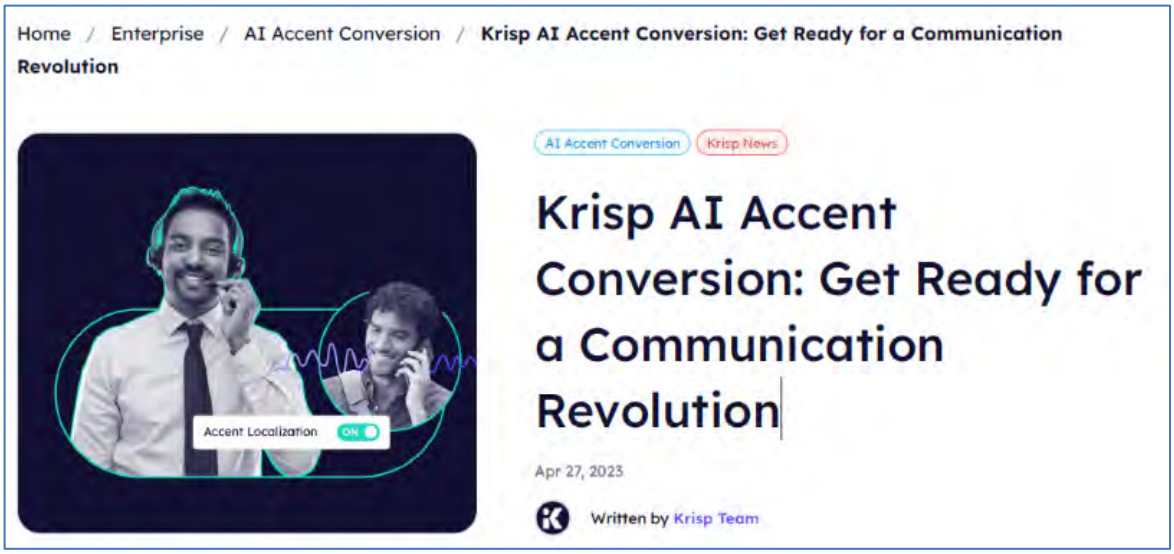
55. At the end of October 2022, Krisp went cold. Sanas' senior management attempted to reach out to their Krisp counterparts but received only silence.

56. Then, without warning, Krisp terminated the discussions. In an email dated November 4, 2022, Krisp's Schoenfield emailed Sanas' Keshava and said that Krisp did not want to proceed with the partnership. He blamed an alleged lack of "visibility" into Sanas' technology—a contention squarely at odds with his engineering team's deep access to Sanas' product. But he also said that Krisp was not yet interested in pursuing accent translation, saying he hoped the companies' prospects for collaboration might change in the future as "Krisp's roadmap and priorities get more aligned with accent technology."

57. Those representations were untrue.

58. At the same time that Krisp was using the prospect of a business relationship to gain access to Sanas' proprietary technology, the details of its performance, how to implement the accent translation technology in real-world customer environments, and Sanas' positioning in the market, ***Krisp was secretly working on its own competing product, using what it learned from Sanas under NDA.***

59. The first hints of the deception came to light only a few months later on April 27, 2023, when Krisp posted an article on its company blog titled "Krisp AI Accent Conversion: Get Ready for a Communication Revolution":



60. The post makes clear that Schoenfield's excuses were lies. Krisp was not waiting for its "roadmap and priorities get more aligned with accent technology"—Krisp was, by its own admission, developing its own competing accent translation system. The post proudly states: "We are excited to announce early access to our newest product release, **Krisp AI Accent Conversion!**"[6] The blog states expressly that the project was not prospective, but the result of past work by the Krisp team:

> "Now, we are excited to announce the addition of yet another game-changing technology to our offering: AI Accent Conversion. ***Our team at Krisp has been working tirelessly to create a technology that utilizes real-time inflection changes to help customers understand agents better by dynamically changing agents' accents into the customer's natively understood accent.*** This innovative product is designed to create better human-to-human connection and communication effectiveness for customers and call centers that are located outside of the United States in countries such as India and the Philippines."

Id. (emphasis added).

[6] https://krisp.ai/blog/krisp-accent-conversion/

61. And the posting shows that the product already had broad capabilities across a "wide range of Indian dialects" with additional capability for "English-speaking Filipino, South African, and Chinese call center agents" "soon to follow." *Id.*

62. The reality is that Krisp used its purported interest in a collaborative partnership with Sanas—a discussion that spanned almost a year—to access Sanas' proprietary technology, its testing of that technology, and the details about how to most effectively implement it, and then misused that information to develop and launch its own competing product just a few months later.

63. Krisp asserts that its products are the result of its own hard work and innovation. "Every day we solve problems that we have never seen in the past. We solve these problems by working hard, constantly learning, adapting, and in the end - always get things done despite obstacles along our path."[7] But that is not true. Krisp's accent conversion is copied from Sanas, and resulted not from Krisp's hard work, but from Sanas'.

DIVISIONAL ASSIGNMENT

64. This Complaint includes an intellectual property action, which is an excepted category under Civil Local Rule 3-2(c). Consequently, this action is assigned on a district-wide basis.

PATENTS-IN-SUIT

65. This action concerns U.S. Patent Nos. 11,948,550 ("the '550 Patent"), 12,125,496 ("the '496 Patent"), 12,131,745 ("the '745 Patent"), and 11,715,457 ("the '457 Patent") (together, the "Asserted Patents").

U.S. Patent No. 11,948,550

66. Sanas is the lawful owner of all right, title, and interest in the '550 Patent entitled "REAL-TIME ACCENT CONVERSION MODEL," including the right to sue and recover for infringement thereof. The '550 Patent was duly and legally issued on April 2, 2024, naming Maxim Serebryakov and Shawn Zhang as the inventors.

[7] https://krisp.ai/about-us/

67. The ’550 Patent has 22 claims: 3 independent claims and 19 dependent claims.

68. The ’550 Patent describes and claims Sanas’ pathbreaking approach to producing high quality, high accuracy, low-latency accent translation that has created a new and valuable commercial market. Sanas’ solution produces remarkably natural sounding outputs, which preserve the speaker’s voice and personality while converting only the speaker’s accent to a more intelligible form. Sanas achieves this high-quality conversion in real-time, thereby allowing for natural conversation without lags or dropped content—all of which are critical for customer-facing applications such as call centers. Sanas delivers these valuable features even while the software runs on a local computer—i.e., without reliance on streaming or a remote computer with the massive computing resources typically used for machine learning models. The ’550 Patent describes particular methods and systems which can be used to achieve the foregoing benefits.

69. A true and correct copy of the ’550 Patent is attached as **Exhibit A**.

U.S. Patent No. 12,125,496

70. Sanas is the lawful owner of all right, title, and interest in the ’496 Patent entitled “METHODS FOR NEURAL NETWORK-BASED VOICE ENHANCEMENT AND SYSTEMS THEREOF,” including the right to sue and recover for infringement thereof. The ’496 Patent was duly and legally issued on October 22, 2024, naming Shawn Zhang, Lukas Pfeifenberger, Jason Wu, Piotr Dura, David Braude, Bajibabu Bollepalli, Alvaro Escudero, Gokce Keskin, Ankita Jha, and Maxim Serebryakov as the inventors.

71. The ’496 Patent has 20 claims: 3 independent claims and 17 dependent claims.

72. The ’496 Patent describes and claims a novel approach to enhancing the intelligibility and quality of voice communication by, for example, removing background noise. The inventions of the ’496 Patent address a particularly difficult problem with creating a functional and effective accent translation system. An accented voice communication, for example in a call center, may include elements that need to be excluded in order for the outputted, converted speech to be intelligible. Those elements could also confuse the artificial intelligence in performing the conversion—thereby potentially jumbling the converted voice. The ’496 Patent describes inventions which include using a low-dimensional representation of input speech frames

to generate target speech frames and ultimately target audio—an approach which enhances the quality and clarity of the output speech. Using this approach produces converted accent audio which is distinct and easily comprehended.

73. A true and correct copy of the '496 Patent is attached as **Exhibit B.**

U.S. Patent No. 12,131,745

74. Sanas is the lawful owner of all right, title, and interest in the '745 Patent entitled "SYSTEM AND METHOD FOR AUTOMATIC ALIGNMENT OF PHONETIC CONTENT FOR REAL-TIME ACCENT CONVERSION," including the right to sue and recover for infringement thereof. The '745 Patent was duly and legally issued on October 29, 2024, naming Lukas Pfeifenberger and Shawn Zhang as the inventors.

75. The '745 Patent has 20 claims: 3 independent claims and 17 dependent claims.

76. The '745 Patent discloses Sanas' solution to another key problem in deploying a commercially valuable implementation of real-time accent translation: how to align phonetically dissimilar audio of two distinct accents. Sub-optimal alignment can lead to poor accent translation accuracy as well as unstable, unintelligible, and/or unnatural sounding speech. It may also provide poor conversions when the source accent is complex and differs substantially from the accents on which the machine learning model was trained. The '745 Patent provides a solution to the alignment problem that produces remarkably smooth, natural-sounding, and accurate accent translated speech.

77. A true and correct copy of the '745 Patent is attached as **Exhibit C.**

U.S. Patent No. 11,715,457

78. Sanas is the lawful owner of all right, title, and interest in the '457 Patent entitled "REAL TIME CORRECTION OF ACCENT IN SPEECH AUDIO SIGNALS", including the right to sue and recover for infringement thereof. The '457 Patent was duly and legally issued on August 1, 2023, naming Andrei Golman and Dmitrii Sadykov as the inventors and Intone Inc. as its assignee. Subsequently, on January 14, 2025, Intone assigned the '457 Patent to Sanas.

79. The '457 Patent has 20 claims: 3 independent claims and 17 dependent claims.

80. A true and correct copy of the '457 Patent is attached as **Exhibit D.**

81. Sanas is the owner of record of each of the Asserted Patents and owns all rights in each of them, including without limitation all rights to recover for past infringement thereof.

82. Sanas has been in compliance with the marking provisions of 35 U.S.C. § 287(a).

KRISP'S WILLFUL INFRINGEMENT OF SANAS' PATENTS

83. Krisp markets and sells a product called "AI Accent Conversion," and has also marketed a product called "Accent Localization." On information and belief, these products, along with prior and subsequent versions (collectively, "the Accused Products") infringe the Sanas patents asserted herein. Krisp's infringement includes the making, using, selling, offering for sale the listed products, as well as Krisp's active inducement of infringement and contributory infringement, including by supplying the listed products to third parties that use those products to practice the claimed methods of the Asserted Patents and that make and use the claimed systems and apparatuses of the Asserted Patents. Sanas reserves the right to supplement and amend its identification of the Accused Products as permitted by the Court.

84. Krisp infringes and continues to infringe the Asserted Patents by making, using, selling, offering to sell, and/or importing, without license or authority, the Accused Products as alleged herein. For example, Krisp advertises, offers for sale, and otherwise promotes the Accused Products on its website. Therein, Krisp describes and touts the use of the subject matter claimed in the Asserted Patents, as described and alleged below. Krisp has also made, used, sold, and offered for sale its products in the United States in connection with its marketing and demonstration of the Accused Products via direct customer marketing in the United States, and marketing at trade shows including the Customer Contact Week (CCW) trade show held annually in Las Vegas, Nevada. Moreover, on information and belief, Krisp has offered for sale and sold the Accused Products to major customers in the United States.

85. Krisp markets, advertises, offers for sale, and/or otherwise promotes the Accused Products and does so to induce, encourage, instruct, and aid one or more persons in the United States to make, use, sell, and/or offer to sell their Accused Products.

86. On information and belief, Krisp has had knowledge of the '550, '496, '745, and '457 Patents since at or around the time of each patent or its parent application being published.

This knowledge is reflected in the fact that such publications are cited in Krisp's own accent conversion patents—U.S. Patent Nos. 12,205,609 ("the '609 Patent") and 12,223,979 ("the '979 Patent"). Those patents both cite U.S. Patent Application 2022/0358903 (which is a parent application for the '550 Patent); and U.S. Patent Application 2024/0347070 (which is a parent application for the '745 Patent). The Krisp patents also cite Sanas' asserted '496 and '457 Patents.

87. In addition, on information and belief, Krisp has paid close attention to Sanas' and Intone's patenting practices and routinely reviews published applications and patents from those companies. As alleged herein, Krisp expressed deep interest in Sanas' accent translation technology as early as 2021 and acknowledged Sanas' leading position in innovating in this space. When the Krisp-Sanas discussions broke off, Krisp told Sanas that the companies were in direct competition. This suggests a systematic effort to assemble competitive intelligence on Sanas, its products, its customers, its market behavior, and its patents.

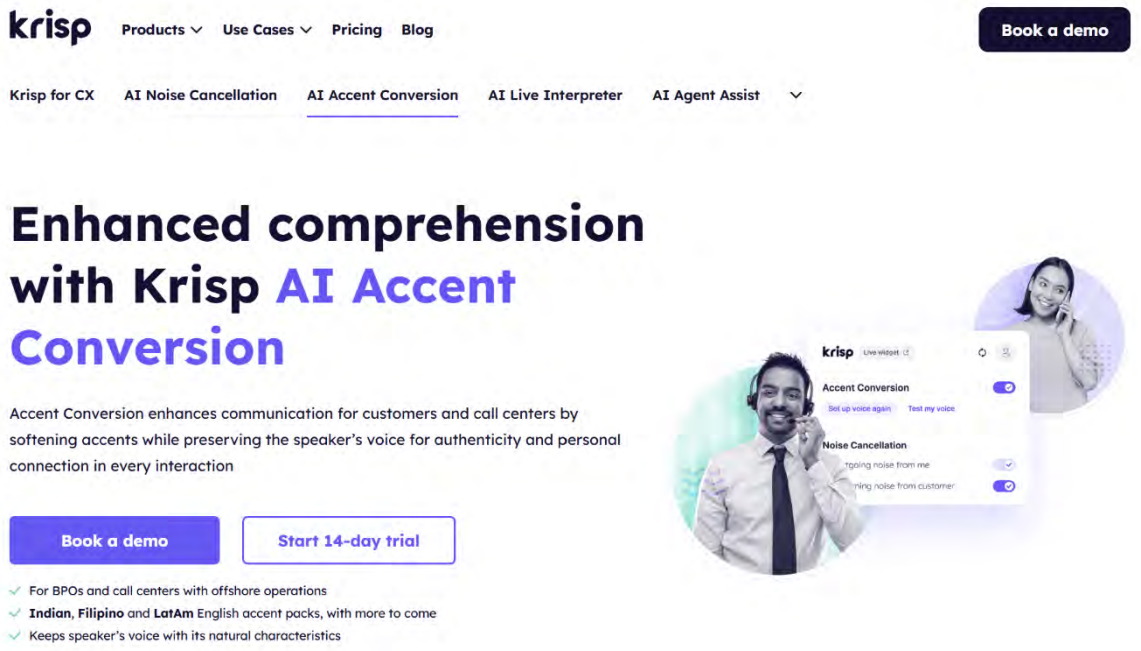
88. On information and belief, Krisp has made, used, sold, or offered for sale the Accused Products with full knowledge of those patents as described above, and knowing that such conduct would constitute an infringement. Krisp acknowledged that Sanas was the leader in development of accent translation technology and affirmatively sought to license Sanas' technology or otherwise form a partnership. But after learning of Sanas' market-ready solutions, Krisp elected to copy Sanas instead. Krisp's imitation rises to the level of willful patent infringement of Sanas' patents and willful misappropriation of Sanas' trade secrets.

COUNT ONE: INFRINGEMENT OF U.S. PATENT 11,948,550

89. Sanas repeats and realleges all preceding paragraphs of this Complaint, as if set forth herein in full.

90. On information and belief, Krisp directly infringes, either literally or under the doctrine of equivalents, at least claim(s) 1-4, and 6-22, of the '550 Patent by making, using, offering for sale, and selling the Accused Products in violation of 35 U.S.C. § 271(a).

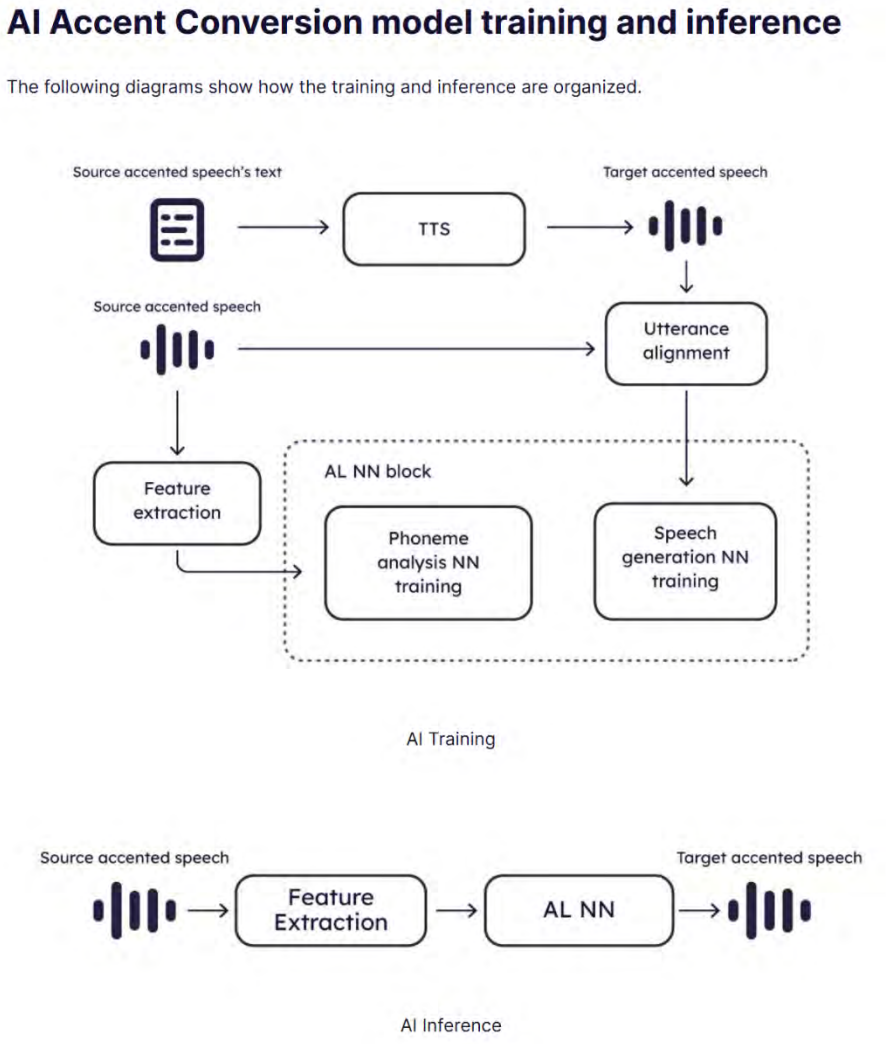
91. For example, Krisp's website describes this product as follows: "Accent Conversion enhances communication for customers and call centers by softening accents while preserving the speaker's voice for authenticity and personal connection in every interaction."



92. A Krisp blog[8] indicates that Krisp's technical approach uses "high-quality parallel data" to "directly map input accented speech to target native speech" such that it "generat[es] a native target-accent sounding output for each accented speech input, maintaining consistent emotions, naturalness, and vocal characteristics, and achieving an ideal frame-by-frame alignment with the input data." A video demo of Krisp's software[9] in use similarly promotes the software as keeping the speaker's "voice intact while softening challenging parts of my accent" and preserving their "natural tone and personality."

[8] https://krisp.ai/blog/deep-dive-ai-accent-conversion-for-call-centers/
[9] www.youtube.com/shorts/uV3xJ5-Sna8

93. The Krisp blog further describes Krisp's technical approach, which includes (among other things) "utterance alignment" and "feature extraction":



94. Krisp has also filed multiple patent applications related to its accent translation product, which have resulted in the issuance of Krisp's two U.S. translation patents, the '609 Patent and the '979 Patent. On information and belief, the written descriptions and/or claims of these patents describe the technical approach employed in the Accused Products.

95. For example, the '979 Patent, along with the web pages, blog post, and demo videos described above, demonstrate that each of the Accused Products is a system that contains all of the elements of claim 1 of the '550 Patent.

96. With knowledge of the '550 Patent, Krisp has also actively induced the infringement of one or more claims of the '550 Patent in violation of 35 U.S.C. § 271(b) by its

customers and/or end users of their products, including at least the Accused Products, by selling products with a particular design, providing for support for, providing instructions for use of, and/or otherwise encouraging its customers and/or end-users to directly infringe, either literally and/or under the doctrine of equivalents, one or more claims of the ’550 Patent, including claim(s) 1-4, and 6-22 , with intent to encourage those customers and/or end-users to infringe the ’550 Patent.

97. By way of example, Krisp has actively induced infringement of the ’550 Patent by encouraging, instructing, and aiding one or more persons in the United States, including but not limited to customers and end users who purchase, test, operate, and use the Accused Products, to make, use, sell, and/or offer to sell Krisp’s products, including the Accused Products, in a manner that infringes at least one claim of the ’550 Patent, including claim(s) 1-4, and 6-22.

98. With knowledge of the ’550 Patent, Krisp has also contributed to the infringement of one or more claims of the ’550 Patent in violation of 35 U.S.C. § 271(c) by its customers and/or end users of their products, including at least the Accused Products, by offering to sell and selling software that constitutes a component of the infringing systems and computer-readable media claimed by the ’550 Patent, and/or a material for use in practicing the methods claimed by the ’550 Patent, which constitutes a material part of the inventions and is not a staple article or commodity of commerce suitable for non-infringing uses. In doing so, Krisp knew that the software would contribute to infringement of the ’550 Patent.

99. With knowledge of the ’550 Patent, Krisp has willfully, deliberately, and intentionally infringed the ’550 Patent. Krisp had actual knowledge of the ’550 Patent and Krisp’s infringement of the ’550 Patent as set forth above. After acquiring that knowledge, Krisp directly and indirectly infringed the ’550 Patent as set forth above. Krisp knew, or should have known, that its conduct amounted to infringement of the ’550 Patent at least because Krisp had sought out technical information from Sanas about the state of its development of Sanas’ products; on information and belief, Krisp was reviewing Sanas’ patent applications and patents in order to build a software product based on Sanas’ innovations; and Krisp’s own products, as reflected in its patents, closely follow the Sanas template for accent translation.

100. Krisp will continue to infringe the ’550 Patent unless it is enjoined by this Court. Krisp, by way of its infringing activities, has caused and continues to cause Sanas to suffer damages in an amount to be determined, and has caused and is causing Sanas irreparable harm. Sanas has no adequate remedy at law against Krisp’s acts of infringement and, unless enjoined from its infringement of the ’550 Patent, Sanas will continue to suffer irreparable harm.

101. Sanas is entitled to recover from Krisp damages at least in an amount adequate to compensate for its infringement of the ’550 Patent, which amount has yet to be determined, together with interest and costs determined by the Court.

102. Sanas has complied with the requirements of 35 U.S.C. § 287 with respect to the ’550 Patent.

COUNT TWO: INFRINGEMENT OF U.S. PATENT 12,125,496

103. Sanas repeats and realleges all preceding paragraphs of this Complaint, as if set forth herein in full.

104. On information and belief, Krisp directly infringes at least claim(s) 1- 20 of the ’496 Patent by making, using, offering for sale, and selling the Accused Products in violation of 35 U.S.C. § 271(a).

105. Krisp’s infringement of the ’496 Patent is reflected in the web pages, blog posts, demo videos, and patents previously mentioned.

106. A Krisp blog post asserts that “[t]he speech synthesis part of the model, which is sometimes referred to as the vocoder algorithm in research, should . . . be robust against noise and background voices,”[10] suggesting that the Accused Products have these characteristics.

107. Krisp has also represented that the Accused Products provide “Background noise and voice cancellation robustness” which is “highly robust, automatically included in the Accent Conversion models.”[11] Krisp further asserts that “Krisp maintains speech quality in real-world noisy conditions, including multi-speaker and contact center environments.”

[10] https://krisp.ai/blog/deep-dive-ai-accent-conversion-for-call-centers/

[11] https://krisp.ai/blog/krisp-vs-sanas-accent-conversion-comparison/. The “comparison” post by Krisp contains numerous falsehoods and radically distorts the relative performance of Krisp’s accent conversion product to Sanas’. The reality is that Sanas has long been the market leader,

108. For example, each of the Accused Products, when installed on a computer, constitutes a voice enhancement system that contains all of the elements of claim 1 of the '496 Patent.

109. With knowledge of the '496 Patent, Krisp has actively induced the infringement of one or more claims of the '496 Patent, in violation of 35 U.S.C. § 271(b), by its customers and/or end users of their products, including at least the Accused Products, by selling products with a particular design, providing for support for, providing instructions for use of, and/or otherwise encouraging its customers and/or end-users to directly infringe, either literally and/or under the doctrine of equivalents, one or more claims of the '496 Patent, including claim(s) 1- 20, with intent to encourage those customers and/or end-users to infringe the '496 Patent.

110. By way of example, Krisp has actively induced infringement of the '496 Patent by encouraging, instructing, and aiding one or more persons in the United States, including but not limited to customers and end users who purchase, test, operate, and use the Accused Products, to make, use, sell, and/or offer to sell Krisp's products, including the Accused Products, in a manner that infringes at least one claim of the '496 Patent, including claim(s) 1-20.

111. With knowledge of the '496 Patent, Krisp has also contributed to the infringement of one or more claims of the '496 Patent in violation of 35 U.S.C. § 271(c) by its customers and/or end users of their products, including at least the Accused Products, by offering to sell and selling software that constitutes a component of the infringing systems and computer-readable media claimed by the '496 Patent, and/or a material for use in practicing the methods claimed by the '496 Patent, which constitutes a material part of the inventions and is not a staple article or commodity of commerce suitable for non-infringing uses. In doing so, Krisp knew that the software would contribute to infringement of the '496 Patent.

112. With knowledge of the '496 Patent, Krisp has willfully, deliberately, and intentionally infringed the '496 Patent. Krisp had actual knowledge of the '496 Patent and Krisp's infringement of the '496 Patent as set forth above. After acquiring that knowledge, Krisp directly

---

and even with Krisp's brazen misappropriation of Sanas' intellectual property, Krisp's product is manifestly inferior to Sanas'.

and indirectly infringed the '496 Patent as set forth above. Krisp knew or should have known that its conduct amounted to infringement of the '496 Patent at least because Krisp had sought out technical information from Sanas about the state of its development of Sanas' products; on information and belief, Krisp was reviewing Sanas' patent applications and patents in order to build a software product based on Sanas' innovations; and Krisp's own products, as reflected in its patents, closely follow the Sanas template for accent translation.

113. Krisp will continue to infringe the '496 Patent unless it is enjoined by this Court. Krisp, by way of its infringing activities, has caused and continues to cause Sanas to suffer damages in an amount to be determined, and has caused, and is causing, Sanas irreparable harm. Sanas has no adequate remedy at law against Krisp's acts of infringement and, unless enjoined from its infringement of the '496 Patent, Sanas will continue to suffer irreparable harm.

114. Sanas is entitled to recover from Krisp damages at least in an amount adequate to compensate for its infringement of the '496 Patent, which amount has yet to be determined, together with interest and costs determined by the Court.

115. Sanas has complied with the requirements of 35 U.S.C. § 287 with respect to the '496 Patent.

COUNT THREE: INFRINGEMENT OF U.S. PATENT 12,131,745

116. Sanas repeats and realleges all preceding paragraphs of this Complaint, as if set forth herein in full.

117. Krisp directly infringes at least claim(s) 1-20 of the '745 Patent by making, using, offering for sale, and selling the Accused Products in violation of 35 U.S.C. § 271(a).

118. For example, a Krisp blog[12] explains that "getting precise alignment is exceedingly challenging due to variations in the duration of phoneme pronunciations. Nonetheless, improved alignment accuracy contributes to superior results." The blog further suggests that Krisp's software "generat[es] a native target-accent sounding output for each accented speech input, maintaining consistent emotions, naturalness, and vocal characteristics, and achieving an ideal

[12] https://krisp.ai/blog/deep-dive-ai-accent-conversion-for-call-centers/

frame-by-frame alignment with the input data." The blog also includes a block diagram which includes a step of "utterance alignment."

119. Krisp's '979 Patent also describes Krisp's alignment process, providing further indication of Krisp's infringement of the '745 Patent.

120. Finally, Krisp's website and other public materials include demonstrations of the output speech produced by the Accused Products. An evaluation of that output speech suggests that the Accused Products do not use non-infringing approaches to alignment of the speech and instead uses an infringing approach including differentiable alignment by maximization of cosine distance between phonetic embedding vectors.

121. Together, the blog, the '979 Patent, and the information available on Krisp's website (including the demonstrations of output speech), reflect that each of the Accused Products when installed on a computer is, on information and belief, an accent translation system containing all of the elements claimed in Claim 1 of the '745 Patent.

122. With knowledge of the '745 Patent, Krisp has actively induced the infringement of one or more claims of the '745 Patent in violation of 35 U.S.C. § 271(b) by its customers and/or end users of their products, including at least the Accused Products, by selling products with a particular design, providing for support for, providing instructions for use of, and/or otherwise encouraging its customers and/or end-users to directly infringe, either literally and/or under the doctrine of equivalents, one or more claims of the '745 Patent, including claim(s) 1-20, with intent to encourage those customers and/or end-users to infringe the '745 Patent.

123. By way of example, Krisp has actively induced infringement of the '745 Patent by encouraging, instructing, and aiding one or more persons in the United States, including but not limited to customers and end users who purchase, test, operate, and use the Accused Products, to make, use, sell, and/or offer to sell Krisp's products, including the Accused Products, in a manner that infringes at least one claim of the '745 Patent, including claim(s) 1-20.

124. With knowledge of the '745 Patent, Krisp has also contributed to the infringement of one or more claims of the '745 Patent in violation of 35 U.S.C. § 271(c) by its customers and/or end users of their products, including at least the Accused Products, by offering to sell and selling

software that constitutes a component of the infringing systems and computer-readable media claimed by the '745 Patent, and/or a material for use in practicing the methods claimed by the '745 Patent, which constitutes a material part of the inventions and is not a staple article or commodity of commerce suitable for non-infringing uses. In doing so, Krisp knew that the software would contribute to infringement of the '745 Patent.

125. With knowledge of the '745 Patent, Krisp has willfully, deliberately, and intentionally infringed the '745 Patent. Krisp had actual knowledge of the '745 Patent and Krisp's infringement of the '745 Patent as set forth above. After acquiring that knowledge, Krisp directly and indirectly infringed the '745 Patent as set forth above. Krisp knew or should have known that its conduct amounted to infringement of the '745 Patent at least because Krisp had sought out technical information from Sanas about the state of its development of Sanas' products; on information and belief, Krisp was reviewing Sanas' patent applications and patents in order to build a software product based on Sanas' innovations; and Krisp's own products, as reflected in its patents, closely follow the Sanas template for accent translation.

126. Krisp will continue to infringe the '745 Patent unless it is enjoined by this Court. Krisp, by way of its infringing activities, has caused and continues to cause Sanas to suffer damages in an amount to be determined, and has caused and is causing Sanas irreparable harm. Sanas has no adequate remedy at law against Krisp's acts of infringement and, unless enjoined from its infringement of the '745 Patent, Sanas will continue to suffer irreparable harm.

127. Sanas is entitled to recover from Krisp damages at least in an amount adequate to compensate for its infringement of the '745 Patent, which amount has yet to be determined, together with interest and costs determined by the Court.

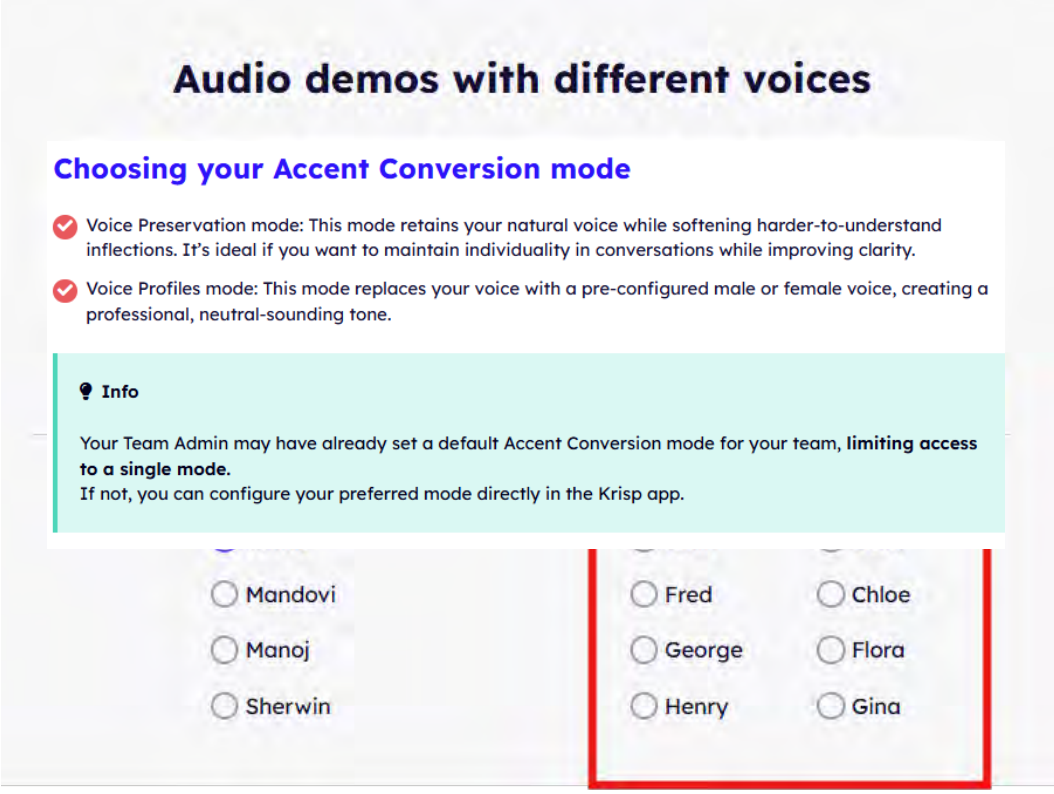
128. Sanas has complied with the requirements of 35 U.S.C. § 287 with respect to the '745 Patent.

COUNT FOUR: INFRINGEMENT OF U.S. PATENT 11,715,457

129. Sanas repeats and realleges all preceding paragraphs of this Complaint, as if set forth herein in full.

130. Krisp directly infringes at least claim(s) 1-20 of the '457 Patent by making, using, offering for sale, and selling the Accused Products in violation of 35 U.S.C. § 271(a).

131. Krisp's website includes audio demos of the performance of the Accused Products, which in turn display the availability of "Voice Profiles" in the Accused Products, which are features that are described and claimed in the '457 Patent.



132. The Accused Products' Voice Profiles Mode is likewise described in a Krisp user guide[13]:

133. Krisp has also claimed in a blog post[14] that the Accused Products have an audio latency of 220ms, as claimed in the '457 Patent.

134. The Krisp blog, websites, demonstrations, and patents described above disclose that each of the Accused Products when installed on a computer is, on information and belief, a computing apparatus containing each of the elements of claim 14 of the '457 Patent.

[13] https://help.krisp.ai/hc/en-us/articles/18308013509916-Krisp-Accent-Conversion-user-guide
[14] https://krisp.ai/blog/krisp-vs-sanas-accent-conversion-comparison/

135. With knowledge of the ’457 Patent, Krisp has actively induced the infringement of one or more claims of the ’457 Patent in violation of 35 U.S.C. § 271(b) by its customers and/or end users of their products, including at least the Accused Products, by selling products with a particular design, providing for support for, providing instructions for use of, and/or otherwise encouraging its customers and/or end-users to directly infringe, either literally and/or under the doctrine of equivalents, one or more claims of the ’457 Patent, including claim(s) 1-20, with intent to encourage those customers and/or end-users to infringe the ’457 Patent.

136. By way of example, Krisp has actively induced infringement of the ’457 Patent by encouraging, instructing, and aiding one or more persons in the United States, including but not limited to customers and end users who purchase, test, operate, and use the Accused Products, to make, use, sell, and/or offer to sell Krisp’s products, including the Accused Products, in a manner that infringes at least one claim of the ’457 Patent, including claim(s) 1-20.

137. With knowledge of the ’457 Patent, Krisp has also contributed to the infringement of one or more claims of the ’457 Patent in violation of 35 U.S.C. § 271(c) by its customers and/or end users of their products, including at least the Accused Products, by offering to sell and selling software that constitutes a component of the infringing computing apparatuses and computer-readable storage media claimed by the ’457 Patent, and/or a material for use in practicing the methods claimed by the ’457 Patent, which constitutes a material part of the inventions and is not a staple article or commodity of commerce suitable for non-infringing uses. In doing so, Krisp knew that the software would contribute to infringement of the ’457 Patent.

138. With knowledge of the ’457 Patent, Krisp has willfully, deliberately, and intentionally infringed the ’457 Patent. Krisp had actual knowledge of the ’457 Patent and Krisp’s infringement of the ’457 Patent as set forth above. After acquiring that knowledge, Krisp directly and indirectly infringed the ’457 Patent as set forth above. Krisp knew or should have known that its conduct amounted to infringement of the ’457 Patent at least because on information and belief, Krisp was reviewing Intone’s patent applications and patents in order to build a software product to compete in the accent translation marketplace.

139. Krisp will continue to infringe the ’457 Patent unless it is enjoined by this Court. Krisp, by way of its infringing activities, has caused and continues to cause Sanas to suffer damages in an amount to be determined, and has caused and is causing Sanas irreparable harm. Sanas has no adequate remedy at law against Krisp’s acts of infringement and, unless enjoined from its infringement of the ’457 Patent, Sanas will continue to suffer irreparable harm.

140. Sanas is entitled to recover from Krisp damages at least in an amount adequate to compensate for its infringement of the ’457 Patent, which amount has yet to be determined, together with interest and costs determined by the Court.

141. Sanas has complied with the requirements of 35 U.S.C. § 287 with respect to the ’457 Patent.

COUNT FIVE: DECLARATORY JUDGMENT

(Co-Inventorship and Co-Ownership Over U.S. Patent Nos. 12,205,609 and 12,223,979)

142. Sanas repeats and realleges all preceding paragraphs of this Complaint, as if set forth herein in full.

143. To the extent they are not invalid, Sanas is at least the joint owner of the ’609 and ’979 patents by virtue of the facts alleged herein, including that during the technical discussions between the parties in 2022, Sanas employees, including Maxim Serebryakov, disclosed to Krisp that Sanas’ accent translation software was being developed based upon a teacher-student architecture; Sanas’ product runs locally on a personal computer CPU; and Sanas’ product uses a parallel speech data processing model.

144. The ’609 and ’979 Patents both incorporate these concepts as central features of the claimed inventions therein. For example, patents (which share a common specification) summarize the inventions in the Abstract as claiming “[t]echniques . . . for generating parallel data for real-time speech form conversion” which include “training a teacher machine learning model that is offline and is substantially larger than a student machine learning model for converting speech form” and “[t]ransferring ‘knowledge’ from the trained Teacher model for training the Production Student Model that performs the speech form conversion on an end-user computing device.” Independent claims 1, 13, and 19 of the ’609 patent include the limitations

of a "teacher machine learning (ML) model;" "parallel speech data;" and "a student machine learning algorithm." The claims of the '979 Patent likewise encompass these concepts.

145. The inventive contributions of one or more Sanas employees, including Mr. Serebryakov, to the subject matter claimed in the '609 and '979 Patents require that they be named as joint inventors of such patents.

146. Sanas' property interests in the '609 and '979 Patents will be prejudiced if the issue of ownership is not adjudicated in connection with the instant action. Unless Sanas obtains from this Court a declaratory judgment of its ownership rights, title, and interests in the '609 and '979 Patents, it faces significant harm, as Krisp's assertion of ownership of the patents prevents Sanas from licensing those patents.

147. Krisp's conduct in asserting that it is the sole owner of the '609 and '979 Patents, and not recognizing the Sanas employees' inventive contributions and Sanas' co-ownership, is a direct and proximate cause of Sanas' injury, which would be redressed by the declaratory judgment sought herein.

148. An actual, present, and justiciable controversy has arisen between Sanas and Krisp concerning ownership of the '609 and '979 Patents.

149. Accordingly, Sanas seeks a declaration that the Sanas employees are co-inventors of the '609 and '979 Patents and that Sanas owns a pro rata undivided interest in the '609 and '979 Patents.

COUNT SIX: MISAPPROPRIATION OF TRADE SECRETS

(Violation of the Defend Trade Secrets Act, 18 U.S.C. § 1836)

150. Sanas repeats and realleges all preceding paragraphs of this Complaint, as if set forth herein in full.

151. As set forth above, Defendant has also improperly and without Sanas' consent accessed, acquired, remained in possession of, used, and/or disclosed certain confidential and proprietary information of Sanas constituting "trade secrets" as defined by 18 U.S.C. § 1839(3). These trade secrets, as described above, are related to a product or service that is used in, that has been used in and/or that is intended for use in interstate and/or foreign commerce. Sanas is the

owner of such information. This information is integral to Sanas' speech AI software solutions and services, and is the result of extensive research, development and investment. These trade secrets were developed, compiled, and enhanced over time by Sanas employees.

152. Sanas has taken numerous, reasonable precautions to protect and to maintain the value of its trade secrets. Sanas employees are subject to obligations to maintain the confidentiality of the trade secrets, including pursuant to employee confidentiality agreements. Licensees of Sanas' software are likewise subject to contractual confidentiality obligations. Sanas also maintains IT security practices and processes, which are designed to and do protect Sanas' sensitive information from being accessed by malicious outside parties. Sanas also divulged sensitive and confidential information to Krisp only after the parties signed a non-disclosure agreement, and Sanas reminded Krisp that the parties' technology teams were working together pursuant to that NDA.

153. Sanas' trade secrets derive actual or potential independent economic value from not being generally known to and not being readily ascertainable through proper means by any other person who can obtain economic value from the disclosure or use of the information.

154. Such trade secrets are not accessible to the public and are not generally known within the trade or by special persons who are skilled in the trade, other than by those who are bound to maintain their secrecy and confidentiality.

155. On information and belief, Krisp has used the Sanas trade secrets without express or implied consent from Sanas, while knowing or having reason to know that the trade secrets were acquired under circumstances giving rise to a duty to maintain the secrecy of the trade secret or limit the use of the trade secret.

156. As a result of the above-described misappropriation, Sanas has been harmed, including by enabling Krisp to improve its ability to compete with Sanas to win customers by using improperly obtained Sanas information to inform the development of its own product, to save on research and development costs, and to otherwise unfairly gain a competitive advantage.

157. As a direct and proximate result of Defendant's unlawful, tortious conduct, Sanas has been damaged and Defendant has been unjustly enriched. The damage to Sanas includes the

loss of revenue from Krisp's use of Sanas' own trade secrets to compete with Sanas for business and to offer services based on those trade secrets at a lower price. The unjust enrichment includes the profits Krisp has obtained through its misappropriation of the trade secrets and the value attributed to the misappropriated information, including amounts Defendant saved in research and development costs using the misappropriated information and increased productivity from use of the misappropriated information.

158. Defendant's conduct constitutes willful and malicious misappropriation within the meaning of the DTSA. In wrongfully and intentionally misappropriating Sanas' trade secrets, as outlined above, Defendant has demonstrated specific intent to cause substantial injury or harm to Sanas. As such, Sanas is entitled to an award of exemplary damages as well as an award of its reasonable attorneys' fees pursuant to the DTSA.

159. Unless Defendant is enjoined from misappropriating Sanas' trade secrets, Sanas will suffer irreparable harm for which there is no adequate remedy at law.

COUNT SEVEN: MISAPPROPRIATION OF TRADE SECRETS

(Violation of the California Uniform Trade Secrets Act, Cal. Civ. Code § 3426 *et seq.*)

160. Sanas repeats and realleges all preceding paragraphs of this Complaint, as if set forth herein in full.

161. As set forth above, Defendant has improperly and without Sanas' consent accessed, acquired, remained in possession of, used, and/or disclosed certain confidential and proprietary information of Sanas constituting "trade secrets" as defined by Cal. Civ. Code § 3426 *et seq.* Sanas is the owner of these trade secrets.

162. Sanas has taken reasonable precautions to protect and maintain the value of its trade secrets, including as described above.

163. Sanas' trade secrets derive actual or potential independent economic value from not being generally known to the public or to other persons who can obtain economic value from the disclosure or use of the information.

164. On information and belief, Krisp has used the Sanas trade secrets without express or implied consent from Sanas, while knowing or having reason to know that the trade secrets

were acquired under circumstances giving rise to a duty to maintain the secrecy of the trade secrets or limit the use of the trade secrets.

165. As a result of the above-described misappropriation, Sanas has been harmed, including by enabling Krisp to improve its ability to compete with Sanas to win customers by using improperly obtained Sanas information to inform the development of its own product, to save on research and development costs, and to otherwise unfairly gain a competitive advantage.

166. Sanas has suffered and will continue to suffer damages and irreparable harm as a direct and proximate result of Defendant's misappropriation of Sanas' trade secrets. As a direct and proximate result of Krisp's misappropriation of Sanas' trade secrets, Defendant has been unjustly enriched and Sanas has sustained damages in an amount to be proven at trial.

167. Defendant's conduct is malicious, oppressive, and deceitful, justifying an award of exemplary damages and attorneys' fees recovery.

168. Unless Defendant is enjoined from misappropriating Sanas' trade secrets, Sanas will suffer irreparable harm for which there is no adequate remedy at law.

PRAYER FOR RELIEF

WHEREFORE, Plaintiff respectfully prays for entry of judgment for Sanas and against Krisp and enter the following relief:

For Patent Infringement:

A. A judgement that Krisp has infringed (directly and/or indirectly) one or more claims of the Asserted Patents, namely U.S. Patent Nos. 11,948,550 ("'550 Patent"), 12,125,496 ("'496 Patent"), 12,131,745 ("'745 Patent"), and 11,715,457 ("'457 Patent") and continues to do so with respect to the '550, '496, '745, and '457 Patents;

B. That Sanas recover all damages to which it is entitled under 35 U.S.C. § 284, including its lost profits, but in no event less than a reasonable royalty;

C. That Krisp be permanently enjoined from further infringement of the Asserted Patents;

D. That Sanas, as the prevailing party, shall recover from Krisp all taxable costs of court;

E. That Sanas shall recover from Krisp all pre- and post-judgment interest on the damages award, calculated at the highest interest rates allowed by law;

F. That Sanas shall recover from Krisp an ongoing royalty in an amount to be determined for continued infringement after the date of judgment;

G. That Krisp's conduct was willful and that Sanas should therefore recover treble damages, including attorneys' fees, expenses, and costs incurred in this action, and an action in the damages award pursuant to 35 U.S.C. § 284;

H. That this case is exceptional and that Sanas shall therefore recover its attorneys' fees and other recoverable expenses, under 35 U.S.C. § 285;

I. That Sanas shall recover from Krisp such other and further relief as the Court deems appropriate;

For Declaratory Judgment:

J. Enter declaratory judgment that the Sanas employees are co-inventors of the '609 and '979 Patents and that Sanas holds an undivided pro rata ownership interest in the '609 and '979 Patents.

For Trade Secret Misappropriation:

K. Permanent injunctive relief enjoining Krisp, and each of their respective agents, servants, employees, attorneys, representatives, and all others acting on their behalf or in concert with them:

1. From any further misappropriation of Sanas' trade secrets;
2. From selling or marketing any product or process derived from misappropriation of any Sanas trade secret;
3. From otherwise further accessing, using, or disclosing Sanas' trade secrets;
4. To return and/or destroy all of Sanas' trade secrets, any record or reflection thereof, and any information derived in whole or in part thereof;
5. To place appropriate restrictions on personnel who have been exposed to any of Sanas' trade secrets or information derived therefrom, including any involvement in product development or customer interactions;

6. To identify and destroy any code (including any source code or operating code) that includes, was derived from, or the creation or modification of which was influenced in any way by the improper access to and/or use of Sanas' trade secrets, including any features or aspect thereof whose creation was aided or motivated by Krisp's access to Sanas' trade secrets;
7. From making false or misleading statements complained of herein or otherwise; and
8. Any other injunctive relief deemed appropriate by the Court;

L. Compensation in an amount to be proven at trial, including but not limited to unjust enrichment, actual losses, lost profits, and/or imposition of a reasonable royalty;

M. An order requiring Krisp to account for all gains, profits, and advantages derived from its misappropriation of Sanas' confidential, proprietary, and/or trade secret information;

N. General and special damages according to proof;

O. Compensatory, exemplary, and punitive damages according to proof;

P. Disgorgement of profits;

Q. Restitution;

R. Pre-judgment and post-judgment interest;

S. Costs of suit;

T. Reasonable attorneys' fees and costs incurred in prosecuting this action; and

U. Such other and further relief as the Court deems just and proper.

DEMAND FOR JURY TRIAL

Plaintiff demands a trial by jury on all claims so triable, pursuant to Fed. R. Civ. P. 38(b).

Dated: July 7, 2025

Respectfully submitted,

/s/ Michael Ng

Michael Ng (SBN 237915)
Michael.Ng@kobrekim.com
Daniel Zaheer (SBN 237118)
Daniel.Zaheer@kobrekim.com
KOBRE & KIM LLP
150 California Street, 19th Floor
San Francisco, CA 94111
Telephone: (415) 582-4800
Fax: (415) 582-4811

Attorneys for Plaintiff
SANAS.AI INC.

EXHIBIT A

US011948550B2

(12) **United States Patent**
Serebryakov et al.

(10) **Patent No.: US 11,948,550 B2**
(45) **Date of Patent: Apr. 2, 2024**

(54) **REAL-TIME ACCENT CONVERSION MODEL**

(71) Applicant: **Sanas.ai Inc.**, Pleasanton, CA (US)

(72) Inventors: **Maxim Serebryakov**, Palo Alto, CA (US); **Shawn Zhang**, Pleasanton, CA (US)

(73) Assignee: **SANAS.AI INC.**, Pleasanton, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **17/460,145**

(22) Filed: **Aug. 27, 2021**

(65) **Prior Publication Data**

US 2022/0358903 A1 Nov. 10, 2022

Related U.S. Application Data

(60) Provisional application No. 63/185,345, filed on May 6, 2021.

(51) **Int. Cl.**
G10L 13/02 (2013.01)
G06N 20/20 (2019.01)
(Continued)

(52) **U.S. Cl.**
CPC ***G10L 13/02*** (2013.01); ***G06N 20/20*** (2019.01); ***G10L 15/02*** (2013.01); ***G10L 25/27*** (2013.01); *G10L 2015/022* (2013.01)

(58) **Field of Classification Search**
CPC G10L 13/02; G10L 15/02
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

10,163,451 B2 12/2018 Dirac et al.
10,614,826 B2 4/2020 Huffman et al.
(Continued)

FOREIGN PATENT DOCUMENTS

CN 111462769 A 7/2020
CN 112382267 A 2/2021
(Continued)

OTHER PUBLICATIONS

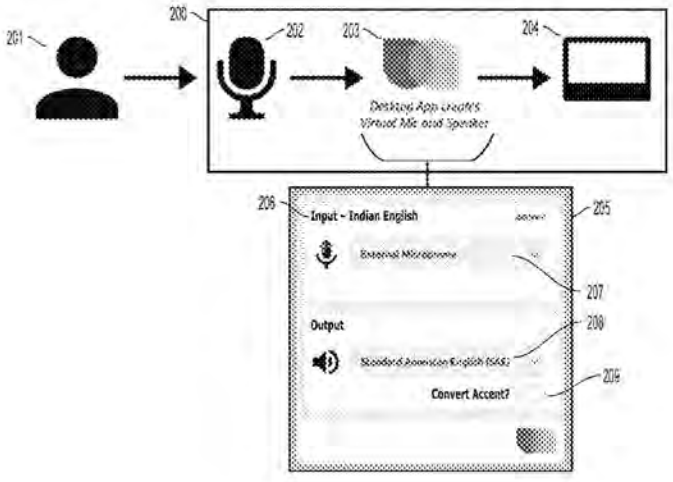
Sajjan, S. C., & Vijaya, C. (Mar. 2016). Continuous Speech Recognition of Kannada language using triphone modeling. In 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET) (pp. 451-455). IEEE.*
(Continued)

Primary Examiner — Bryan S Blankenagel
(74) *Attorney, Agent, or Firm* — Troutman Pepper Hamilton Sanders LLP

(57) **ABSTRACT**

Techniques for real-time accent conversion are described herein. An example computing device receives an indication of a first accent and a second accent. The computing device further receives, via at least one microphone, speech content having the first accent. The computing device is configured to derive, using a first machine-learning algorithm trained with audio data including the first accent, a linguistic representation of the received speech content having the first accent. The computing device is configured to, based on the derived linguistic representation of the received speech content having the first accent, synthesize, using a second machine learning-algorithm trained with (i) audio data comprising the first accent and (ii) audio data including the second accent, audio data representative of the received speech content having the second accent. The computing device is configured to convert the synthesized audio data into a synthesized version of the received speech content having the second accent.

22 Claims, 4 Drawing Sheets



(51) **Int. Cl.**
G10L 15/02 (2006.01)
G10L 25/27 (2013.01)

(56) **References Cited**

U.S. PATENT DOCUMENTS

11,361,753	B2 *	6/2022	Pan	G10L 13/10
2003/0018473	A1 *	1/2003	Ohnishi	G10L 13/10 704/258
2006/0129399	A1 *	6/2006	Turk	G10L 21/00 704/E21.001
2008/0133241	A1 *	6/2008	Baker	G10L 19/0018 704/260
2009/0204395	A1 *	8/2009	Kato	G10L 13/033 704/E11.001
2010/0211376	A1 *	8/2010	Chen	G10L 15/187 704/250
2014/0258462	A1 *	9/2014	Hwang	G06Q 30/0621 709/219
2014/0365216	A1 *	12/2014	Gruber	G10L 13/04 704/235
2015/0170642	A1 *	6/2015	Peng	G10L 15/187 704/235
2016/0203827	A1 *	7/2016	Leff	G06T 13/40 704/207
2020/0169591	A1 *	5/2020	Ingel	G10L 13/08
2021/0074264	A1 *	3/2021	Liang	G06N 20/00
2022/0122579	A1 *	4/2022	Biadsy	G10L 21/003
2022/0148562	A1 *	5/2022	Park	G10L 13/047
2022/0180762	A1 *	6/2022	Lurie	G10L 15/22
2022/0301542	A1 *	9/2022	Sung	G10L 13/02

FOREIGN PATENT DOCUMENTS

CN	112382270 A	2/2021
JP	2005234418 A	9/2005

OTHER PUBLICATIONS

Zhao, G., Ding, S., & Gutierrez-Osuna, R. (2019). Foreign Accent Conversion by Synthesizing Speech from Phonetic Posteriorgrams. In InterSpeech (pp. 2843-2847).*

Zheng, D. C. (2011). Accent Conversion via Formant-based Spectral Mapping and Pitch Contour Modification.*

International Searching Authority, PCT International Search Report and Written Opinion, PCT International Application No. PCT/US2022/028156, dated Aug. 17, 2022, 8 pages.

* cited by examiner

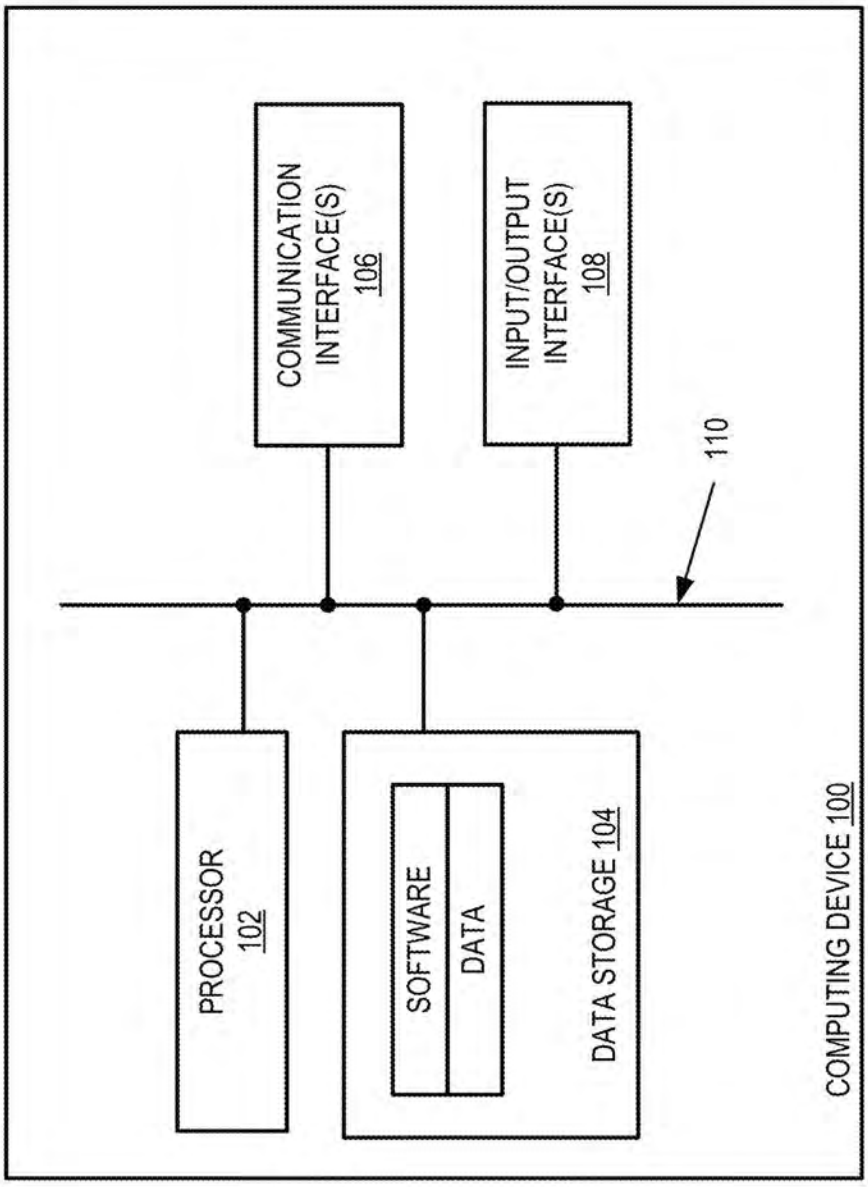

COMMUNICATION INTERFACE(S) 106
INPUT/OUTPUT INTERFACE(S) 108
110
PROCESSOR 102
SOFTWARE
DATA
DATA STORAGE 104
COMPUTING DEVICE 100

FIG. 1

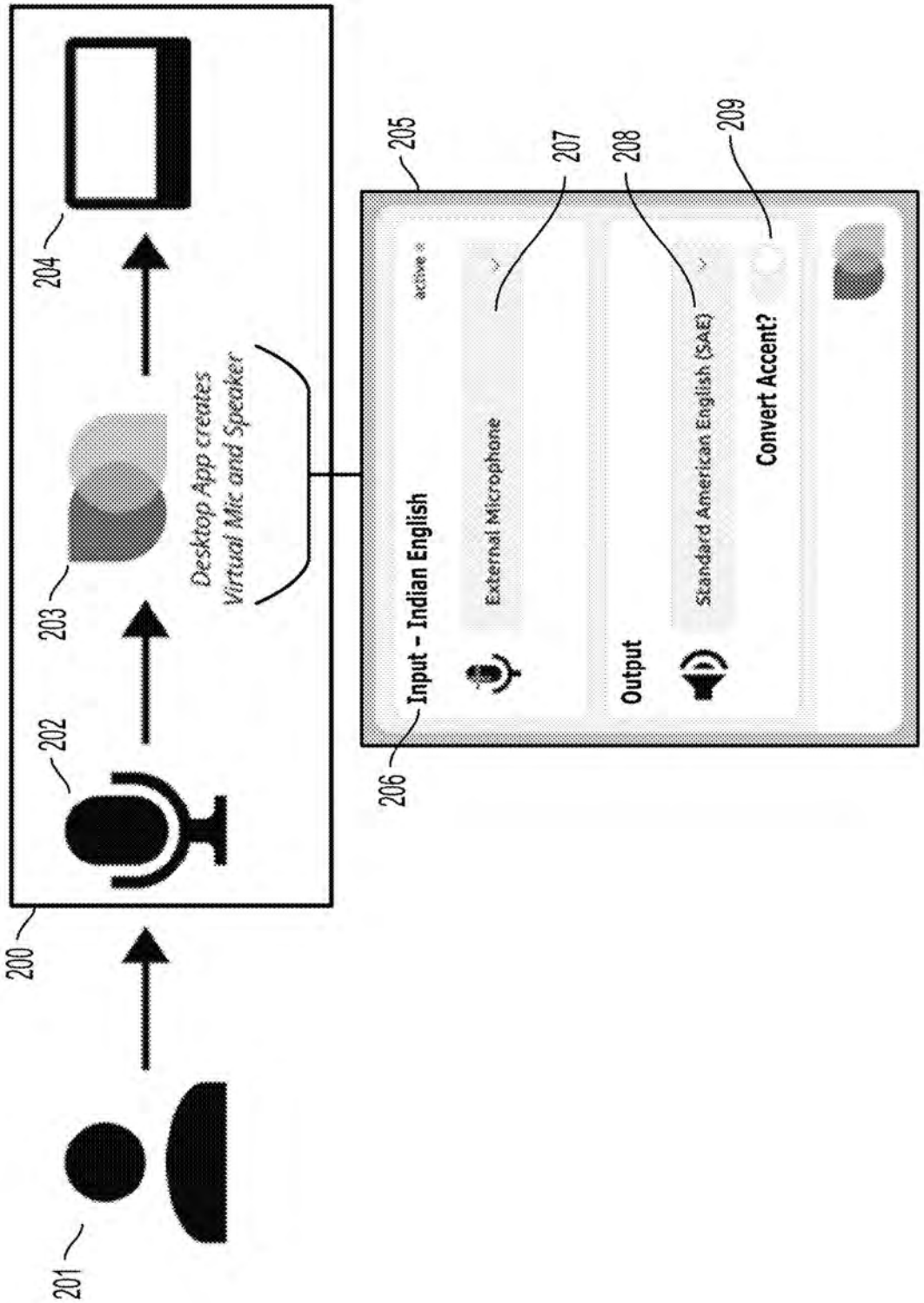

201
200
202
203
204
Desktop App creates
Virtual Mic and Speaker
205
206
207
208
209
Input – Indian English
External Microphone
Output
Standard American English (SAE)
Convert Accent?

FIG. 2

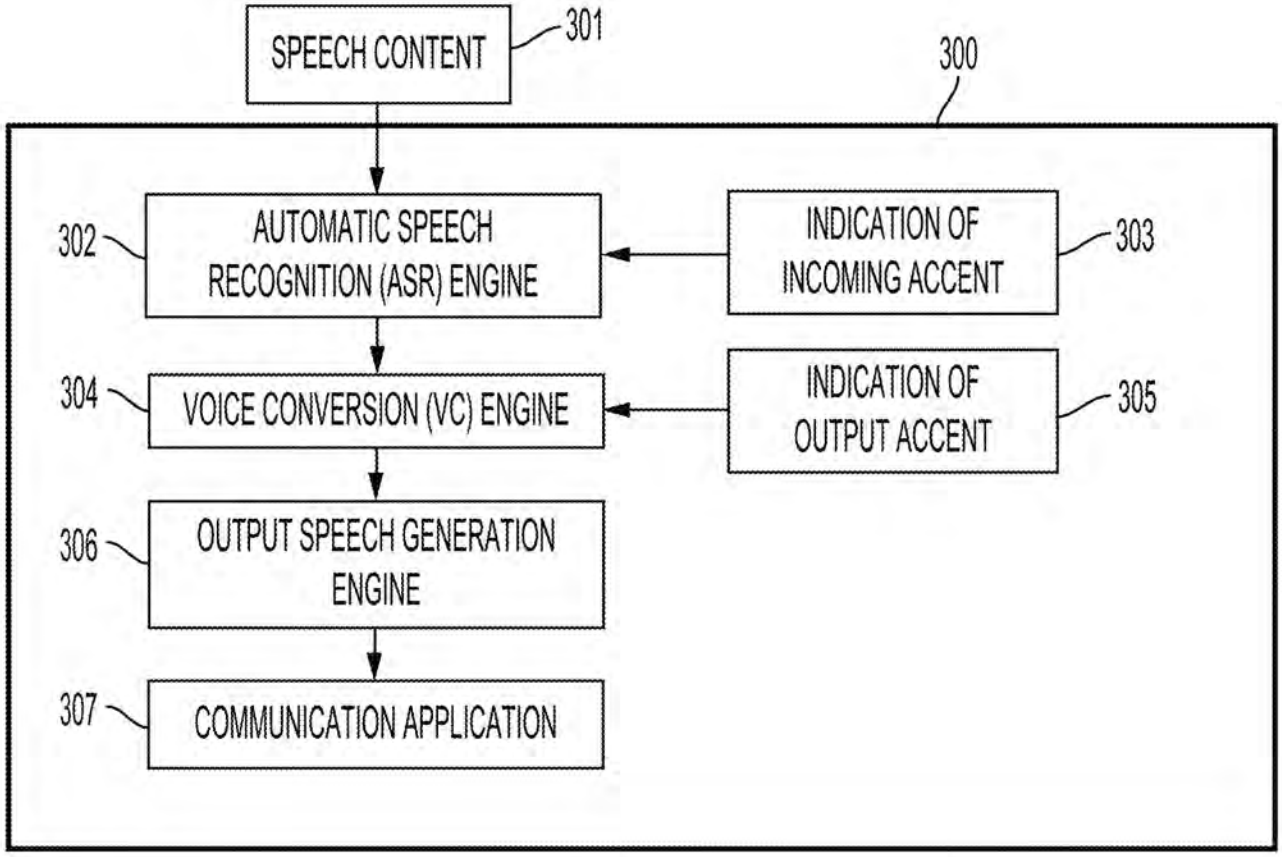

SPEECH CONTENT
301
300
302
AUTOMATIC SPEECH RECOGNITION (ASR) ENGINE
INDICATION OF INCOMING ACCENT
303
304
VOICE CONVERSION (VC) ENGINE
INDICATION OF OUTPUT ACCENT
305
306
OUTPUT SPEECH GENERATION ENGINE
307
COMMUNICATION APPLICATION

FIG. 3

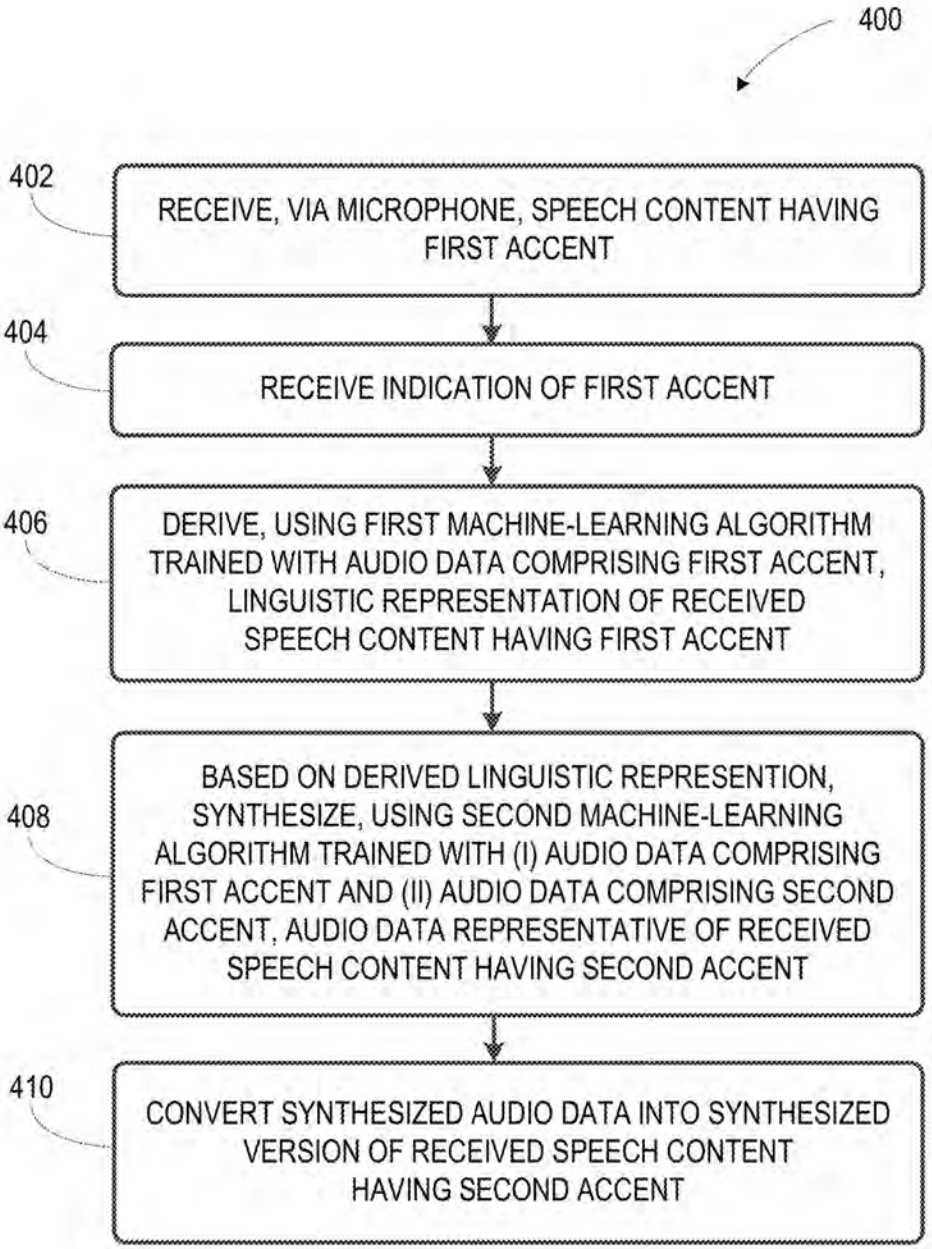

400
402
RECEIVE, VIA MICROPHONE, SPEECH CONTENT HAVING FIRST ACCENT
404
RECEIVE INDICATION OF FIRST ACCENT
406
DERIVE, USING FIRST MACHINE-LEARNING ALGORITHM TRAINED WITH AUDIO DATA COMPRISING FIRST ACCENT, LINGUISTIC REPRESENTATION OF RECEIVED SPEECH CONTENT HAVING FIRST ACCENT
408
BASED ON DERIVED LINGUISTIC REPRESENTION, SYNTHESIZE, USING SECOND MACHINE-LEARNING ALGORITHM TRAINED WITH (I) AUDIO DATA COMPRISING FIRST ACCENT AND (II) AUDIO DATA COMPRISING SECOND ACCENT, AUDIO DATA REPRESENTATIVE OF RECEIVED SPEECH CONTENT HAVING SECOND ACCENT
410
CONVERT SYNTHESIZED AUDIO DATA INTO SYNTHESIZED VERSION OF RECEIVED SPEECH CONTENT HAVING SECOND ACCENT

FIG. 4

REAL-TIME ACCENT CONVERSION MODEL

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims the benefit of priority under 35 U.S.C. § 119 to U.S. Provisional Patent App. No. 63/185,345, filed on May 6, 2021, which is incorporated herein by reference in its entirety.

BACKGROUND

Software applications are used on a regular basis to facilitate communication between users. As some examples, software applications can facilitate text-based communications such as email and other chatting/messaging platforms. Software applications can also facilitate audio and/or video-based communication platforms. Many other types of software applications for facilitating communications between users exist.

Software applications are increasingly being relied on for communications in both personal and professional capacities. It is therefore desirable for software applications to provide sophisticated features and tools which can enhance a user's ability to communicate with others and thereby improve the overall user experience. Thus, any tool that can improve a user's ability to communicate with others is desirable.

OVERVIEW

One of the oldest communication challenges faced by people around the world is the barrier presented by different languages. Further, even among speakers of the same language, accents can sometimes present a communication barrier that is nearly as difficult to overcome as if the speakers were speaking different languages. For instance, a person who speaks English with a German accent may have difficulty understanding a person who speaks English with a Scottish accent.

Today, there are relatively few software-based solutions that attempt to address the problem of accent conversion between speakers of the same language. One type of approach that has been proposed involves using voice conversion methods that attempt to adjust the audio characteristics (e.g., pitch, intonation, melody, stress) of a first speaker's voice to more closely resemble the audio characteristics of a second speaker's voice. However, this type of approach does not account for the different pronunciations of certain sounds that are inherent to a given accent, and therefore these aspects of the accent remain in the output speech. For example, many accents of the English language, such as Indian English and Irish English do not pronounce the phoneme for the digraph "th" found in Standard American English (SAE), instead replacing it with a "d" or "t" sound (sometimes referred to as th-stopping). Accordingly, a voice conversion model that only adjusts the audio characteristics of input speech does not address these types of differences.

Some other approaches have involved a speech-to-text (STT) conversion of input speech as a midpoint, followed by a text-to-speech (TTS) conversion to generate the output audio content. However, this type of STT-TTS approach cannot capture many of the nuances of input speech that can provide information beyond the meaning of the words themselves, such as the prosody or emotion of the speaker. Further, a STT-TTS approach generally involves a degree of latency (e.g., up to several seconds) that makes it impractical for use in real-time communication scenarios such as an ongoing conversation (e.g., a phone call).

To address these and other problems with existing solutions for performing accent conversion, disclosed herein is new software technology that utilizes machine-learning models to receive input speech in a first accent and then output a synthesized version of the input speech in a second accent, all with very low latency (e.g., 300 milliseconds or less). In this way, accent conversion may be performed by a computing device in real time, allowing two users to verbally communicate more effectively in situations where their different accents would have otherwise made such communication difficult.

Accordingly, in one aspect, disclosed herein is a method that involves a computing device (i) receiving an indication of a first accent, (ii) receiving, via at least one microphone, speech content having the first accent, (iii) receiving an indication of a second accent, (iv) deriving, using a first machine-learning algorithm trained with audio data comprising the first accent, a linguistic representation of the received speech content having the first accent, (v) based on the derived linguistic representation of the received speech content having the first accent, synthesizing, using a second machine learning-algorithm trained with (a) audio data comprising the first accent and (b) audio data comprising the second accent, audio data representative of the received speech content having the second accent, and (vi) converting the synthesized audio data into a synthesized version of the received speech content having the second accent.

In another aspect, disclosed herein is a computing device that includes at least one processor, a communication interface, a non-transitory computer-readable medium, and program instructions stored on the non-transitory computer-readable medium that are executable by the at least one processor to cause the computing device to carry out the functions disclosed herein, including but not limited to the functions of the foregoing method.

In yet another aspect, disclosed herein is a non-transitory computer-readable storage medium provisioned with software that is executable to cause a computing device to carry out the functions disclosed herein, including but not limited to the functions of the foregoing method.

One of ordinary skill in the art will appreciate these as well as numerous other aspects in reading the following disclosure.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** depicts an example computing device that may be configured to carry out one or more functions of a real-time accent conversion model.

FIG. **2** depicts a simplified block diagram of a computing device configured for real-time accent conversion.

FIG. **3** depicts a simplified block diagram of a computing device and an example data flow pipeline for a real-time accent conversion model.

FIG. **4** depicts an example flow chart that may be carried out to facilitate using a real-time accent conversion model.

DETAILED DESCRIPTION

The following disclosure refers to the accompanying figures and several example embodiments. One of ordinary skill in the art should understand that such references are for the purpose of explanation only and are therefore not meant to be limiting. Part or all of the disclosed systems, devices,

and methods may be rearranged, combined, added to, and/or removed in a variety of manners, each of which is contemplated herein.

I. Example Computing Device

FIG. **1** is a simplified block diagram illustrating some structural components that may be included in an example computing device **100**, on which the software technology discussed herein may be implemented. As shown in FIG. **1**, the computing device may include one or more processors **102**, data storage **104**, a communication interface **106**, one or more input/output (I/O) interfaces **108**, all of which may be communicatively linked by a communication link **110** that may take the form of a system bus, among other possibilities.

The processor **102** may comprise one or more processor components, such as general-purpose processors (e.g., a single- or multi-core microprocessor), special-purpose processors (e.g., an application-specific integrated circuit or digital-signal processor), programmable logic devices (e.g., a field programmable gate array), controllers (e.g., microcontrollers), and/or any other processor components now known or later developed. In line with the discussion above, it should also be understood that processor **102** could comprise processing components that are distributed across a plurality of physical computing devices connected via a network, such as a computing cluster of a public, private, or hybrid cloud.

In turn, data storage **104** may comprise one or more non-transitory computer-readable storage mediums that are collectively configured to store (i) software components including program instructions that are executable by processor **102** such that computing device **100** is configured to perform some or all of the disclosed functions and (ii) data that may be received, derived, or otherwise stored, for example, in one or more databases, file systems, or the like, by computing device **100** in connection with the disclosed functions. In this respect, the one or more non-transitory computer-readable storage mediums of data storage **104** may take various forms, examples of which may include volatile storage mediums such as random-access memory, registers, cache, etc. and non-volatile storage mediums such as read-only memory, a hard-disk drive, a solid-state drive, flash memory, an optical-storage device, etc. In line with the discussion above, it should also be understood that data storage **104** may comprise computer-readable storage mediums that are distributed across a plurality of physical computing devices connected via a network, such as a storage cluster of a public, private, or hybrid cloud. Data storage **104** may take other forms and/or store data in other manners as well.

The communication interface **106** may be configured to facilitate wireless and/or wired communication between the computing device **100** and other systems or devices. As such, communication interface **106** may communicate according to any of various communication protocols, examples of which may include Ethernet, Wi-Fi, Controller Area Network (CAN) bus, serial bus (e.g., Universal Serial Bus (USB) or Firewire), cellular network, and/or short-range wireless protocols, among other possibilities. In some embodiments, the communication interface **106** may include multiple communication interfaces of different types. Other configurations are possible as well.

The I/O interfaces **108** of computing device **100** may be configured to (i) receive or capture information at computing device **100** and/or (ii) output information for presentation to a user. In this respect, the one or more I/O interfaces **108** may include or provide connectivity to input components such as a microphone, a camera, a keyboard, a mouse, a trackpad, a touchscreen, or a stylus, among other possibilities. Similarly, the I/O interfaces **108** may include or provide connectivity to output components such as a display screen and an audio speaker, among other possibilities.

It should be understood that computing device **100** is one example of a computing device that may be used with the embodiments described herein, and may be representative of the computing devices **200** and **300** shown in FIGS. **2-3** and discussed in the examples below. Numerous other arrangements are also possible and contemplated herein. For instance, other example computing devices may include additional components not pictured or include less than all of the pictured components.

II. Example Functionality

Turning to FIG. **2**, a simplified block diagram of a computing device configured for real-time accent conversion is shown. As described above, the disclosed technology is generally directed to a new software application that utilizes machine-learning models to perform real-time accent conversion on input speech that is received by a computing device, such as the computing device **200** shown in FIG. **2**. In this regard, the accent-conversion application may be utilized in conjunction with one or more other software applications that are normally used for digital communications.

For example, as shown in FIG. **2**, a user **201** of the computing device **200** may provide speech content that is captured by a hardware microphone **202** of the computing device **200**. In some embodiments, the hardware microphone **202** shown in FIG. **2** might be an integrated component of the computing device **200** (e.g., the onboard microphone of a laptop computer or smartphone). In other embodiments, the hardware microphone **202** might take the form of a wired or wireless peripheral device (e.g., a webcam, a dedicated hardware microphone) that is connected to an I/O interface of the computing device **200**. Other examples are also possible.

The speech content may then be passed to the accent-conversion application **203** shown in FIG. **2**. In some implementations, the accent-conversion application **203** may function as a virtual microphone that receives the captured speech content from the hardware microphone **202** of the computing device **200**, performs accent conversion as discussed herein, and then routes the converted speech content to a digital communication application **204** (e.g., a digital communication application such as those using the trade names Zoom®, Skype®, Viber®, Telegram®, etc.) that would normally receive input speech content directly from the hardware microphone **202**. Advantageously, the accent conversion may be accomplished locally on the computing device **200**, which may tend to minimize the latency associated with other applications that may rely on cloud-based computing.

FIG. **2** shows one possible example of a virtual microphone interface **205** that may be presented by the accent-conversion application **203**. For example, the virtual microphone interface **205** may provide an indication **206** of the input accent of the user **201**, which may be established by the user **201** upon initial installation of the accent-conversion application **203** on computing device **200**. As shown in FIG. **2**, the virtual microphone interface **205** indicates that the user **201** speaks with an Indian English accent. In some implementations, the input accent may be adjustable to accommodate users with different accents than the user **201**.

Further, the virtual microphone interface **205** may include a drop-down menu **207** or similar option for selecting the

input source from which the accent-conversion application **203** will receive speech content, as the computing device **200** might have multiple available options to use as an input source. Still further, the virtual microphone interface **205** may include a drop-down menu **208** or similar option for selecting the desired output accent for the speech content. As shown in FIG. **2**, the virtual microphone interface **205** indicates that the incoming speech content will be converted to speech having a SAE accent. The converted speech content is then provided to the communication application **204**, which may process the converted speech content as if it had come from the hardware microphone **202**.

Still further, in some implementations, the virtual microphone interface **205** may include a toggle **209** or similar control that may be used to turn the accent conversion functionality of the accent-conversion application **203** on or off. When the conversion functionality is toggled off, the accent-conversion application **203** may act as a pass-through for the input speech. In this way, the user **201** may avoid the hassle of reconfiguring input devices to remove the virtual microphone for conversations where accent conversion is not needed, allowing the user **201** to easily move between conversations with and without accent conversion engaged.

Advantageously, the accent-conversion application **203** may accomplish the operations above, and discussed in further detail below, at speeds that enable real-time communications, having a latency as low as 50-700 ms (e.g., 200 ms) from the time the input speech received by the accent-conversion application **203** to the time the converted speech content is provided to the communication application **204**. Further, the accent-conversion application **203** may process incoming speech content as it is received, making it capable of handling both extended periods of speech as well as frequent stops and starts that may be associated with some conversations. For example, in some embodiments, the accent-conversion application **203** may process incoming speech content every 160 ms. In other embodiments, the accent-conversion application **203** may process the incoming speech content more frequently (e.g., every 80 ms) or less frequently (e.g., every 300 ms).

Turning now to FIG. **3**, a simplified block diagram of a computing device **300** and an example data flow pipeline for a real-time accent conversion model are shown. For instance, the computing device **300** may be similar to or the same as the computing device **200** shown in FIG. **2**. At a high-level, the components of the real-time accent conversion model that operate on the incoming speech content **301** include (i) an automatic speech recognition (ASR) engine **302**, (ii) a voice conversion (VC) engine **304**, and (iii) an output speech generation engine **306**. As one example, the output speech generation engine may be embodied in a vocoder **306**.

FIG. **3** will be discussed in conjunction with FIG. **4**, which depicts a flow chart **400** that includes example operations that may be carried out by a computing device, such as the computing device **300** of FIG. **3**, to facilitate using a real-time accent conversion model.

At block **402**, the computing device **300** may receive speech content **301** having a first accent. For instance, as discussed above with respect to FIG. **2**, a user such as user **201** may provide speech content **301** having an Indian English accent, which may be captured by a hardware microphone of the computing device **300**. In some implementations, the computing device **300** may engage in pre-processing of the speech content **301**, including converting the speech content **301** from an analog signal to a digital signal using an analog-to-digital converter (not shown), and/or down-sampling the speech content **301** to a sample rate (e.g., 16 kHz) that will be used by the ASR engine **302**, among other possibilities. In other implementations, one or more of these pre-processing actions may be performed by the ASR engine **302**.

The ASR engine **302** includes one or more machine learning models (e.g., a neural network, such as a recurrent neural network (RNN), a transformer neural network, etc.) that may be trained using previously captured speech content from many different speakers having the first accent. Continuing the example above, the ASR engine **302** may be trained with previously captured speech content from a multitude of different speakers, each having an Indian English accent. For instance, the captured speech content used as training data may include transcribed content in which each of the speakers read the same script (e.g., a script curated to provide a wide sampling of speech sounds, as well as specific sounds that are unique to the first accent). Thus, the ASR engine **302** may align and classify each frame of the captured speech content according to its monophone and triphone sounds, as indicated in the corresponding transcript. As a result of this frame-wise breakdown of the captured speech across multiple speakers having the first accent, the ASR engine **302** may develop a learned linguistic representation of speech having an Indian English accent that is not speaker-specific.

On the other hand, the ASR engine **302** may also be used to develop a learned linguistic representation for an output accent that is only based on speech content from a single, representative speaker (e.g., a target SAE speaker) reading a script in the output accent, and therefore is speaker specific. In this way, the synthesized speech content that is generated having the target accent (discussed further below) will tend to sound like the target speaker for the output accent. In some cases, this may simplify the processing required to perform accent conversion and generally reduce latency.

Further, it should be understood that the captured speech content that is used to train the ASR engine **302** does not necessarily need to be limited to captured speech content having the first accent. Rather, the ASR engine **302** discussed herein may be trained using captured speech content having a diversity of accents, which may enable the ASR engine **302** to develop a learned linguistic representation of not only the first accent, but also the second accent as well as additional accents. In this way, the accent-conversion application **203** noted above may utilize a single ASR engine **302** that is capable of receiving and converting speech content having various different input accents.

In some implementations, the speech content collected from the multiple Indian English speakers as well as the target SAE speaker for training the ASR engine **302** may be based on the same script, also known as parallel speech. In this way the transcripts used by the ASR engine **302** to develop a linguistic representation for speech content in both accents are the same, which may facilitate mapping one linguistic representation to the other in some situations. Alternatively, the training data may include non-parallel speech, which may require less training data. Other implementations are also possible, including hybrid parallel and non-parallel approaches.

It should be noted that the learned linguistic representations developed by the ASR engine **302** and discussed herein may not be recognizable as such to a human. Rather, the learned linguistic representations may be encoded as machine-readable data (e.g., a hidden representation) that the ASR engine **302** uses to represent linguistic information.

In practice, the ASR engine **302** may be individually trained with speech content including multiple different accents, across different languages, and may develop a learned linguistic representation for each one. Accordingly, at block **404**, the computing device **300** may receive an indication of the Indian English accent associated with the received speech content **301**, so that the appropriate linguistic representation is used by the ASR engine **302**. As noted above, this indication of the incoming accent, shown by way of example as block **303** in FIG. **3**, may be established at the time the accent-conversion application is installed on the computing device **300** and might not be changed thereafter. As another possibility, the accent-conversion application may be adjusted to indicate a different incoming accent, such that the ASR engine **302** uses a different learned linguistic representation to analyze the incoming speech content **301**.

At block **406**, the ASR engine **302** may derive a linguistic representation of the received speech content **301**, based on the learned linguistic representation the ASR engine **302** has developed for the Indian English accent. For instance, the ASR engine **302** may break down the received speech content **301** by frame and classify each frame according to the sounds (e.g., monophones and triphones) that are detected, and according to how those particular sounds are represented and inter-related in the learned linguistic representation associated with an Indian English accent.

In this way, the ASR engine **302** functions to deconstruct the received speech content **301** having the first accent into a derived linguistic representation with very low latency. In this regard, it should be noted that the ASR engine **302** may differ from some other speech recognition models that are configured predict and generate output speech, such as a speech-to-text model. Accordingly, the ASR engine **302** may not need to include such functionality.

The derived linguistic representation of the received speech content **301** may then be passed to the VC engine **304**. Similar to the indication of the incoming accent **303**, the computing device **300** may also receive an indication of the output accent, shown by way of example as block **305** in FIG. **3**, so that the VC engine **304** can apply the appropriate mapping and conversion from the incoming accent to the output accent. For instance, the indication of the output accent may be received based on a user selection from a menu, such as the virtual microphone interface **205** shown in FIG. **2**, prior to receiving the speech content **301** having the first accent.

Similar to the ASR engine **302**, the VC engine **304** includes one or more machine learning models (e.g., a neural network) that use the learned linguistic representations developed by the ASR engine **302** as training inputs to learn how to map speech content from one accent to another. For instance, the VC engine **304** may be trained to map an ASR-based linguistic representation of Indian English speech to an ASR-based linguistic representation of a target SAE speaker. In training the VC engine **304**, it is necessary to align the Indian English speech to the SAE speech during this mapping. One possible way to accomplish this is by using individual monophones and/or triphones within the training data as a possible heuristic to better determine the alignments. Like the learned linguistic representations themselves, the learned mapping between the two representations may be encoded as machine-readable data (e.g., a hidden representation) that the VC engine **304** uses to represent linguistic information.

Accordingly, at block **408**, the VC engine **304** may utilize the learned mapping between the two linguistic representations to synthesize, based on the derived linguistic representation of the received speech content **301**, audio data that is representative of the speech content **301** having the second accent. The audio data that is synthesized in this way may take the form of a set of mel spectrograms. For example, the VC engine **304** may map each incoming frame in the derived linguistic representation to an outgoing target speech frame.

In this way, the VC engine **304** functions to reconstruct acoustic features from the derived linguistic representation into audio data that is representative of speech by a different speaker having the second accent, all with very low latency. Advantageously, because the VC engine **304** works at the level of encoded linguistic data and does not need to predict and generate output speech as a midpoint for the conversion, it can function more quickly than alternatives such as a STT-TTS approach. Further, the VC engine **304** may more accurately capture some of the nuances of voice communications, such as brief pauses or changes in pitch, prosody, or the emotion of the speaker, all of which can convey important information and which may be lost if the speech content were converted to text first and then back to speech.

At block **410**, the output speech generation engine **306** may convert the synthesized audio data into output speech, which may be a synthesized version of the received speech content **301** having the second accent. As noted above, the output speech may further have the voice identity of the target speaker whose speech content, having the second accent, was used to train the ASR engine **302**. In some examples, the output speech generation engine **306** may take the form of a vocoder **306** or similar component that can rapidly process audio under the real-time conditions contemplated herein. The output speech generation engine **306** may include one or more additional machine learning algorithms (e.g., a neural network, such as a generative adversarial network, one or more Griffin-Lim algorithms, etc.) that learn to convert the synthesized audio data into waveforms that are able to be heard. Other examples are also possible.

As shown in FIG. **3**, the output speech generation engine **306** may pass the output speech to a communication application **307** operating on the computing device **300**. The communication application **307** may then transmit the output speech to one or more other computing devices, cause the computing device **300** to play back the output speech via one or more speakers, and/or store the output speech as an audio data file, among numerous other possibilities.

Although the examples discussed above involve a computing device **300** that utilizes the accent-conversation application for outgoing speech (e.g., situations where the user of computing device **300** is the speaker), it is also contemplated that the accent-conversion application may be used by the computing device **300** in the opposite direction as well, for incoming speech content where the user is a listener. For instance, rather than being situated as a virtual microphone between a hardware microphone and the communication application **307**, the accent-conversion application may be deployed as a virtual speaker between the communication application **307** and a hardware speaker of the computing device **300**, and the indication of the incoming accent **303** and the indication of the output accent **305** shown in FIG. **3** may be swapped. In some cases, these two pipelines may run in parallel such that a single installation of the accent-conversion application is performing two-way accent conversion between users. In the context of the example discussed above, this arrangement may allow the Indian English speaker, whose outgoing speech is being converted to an SAE accent, to also hear the SAE speaker's responses

in Indian English accented speech (e.g., synthesized speech of a target Indian English speaker).

As a further extension, the examples discussed above involve an ASR engine **302** that is provided with an indication of the incoming accent. However, in some embodiments it may be possible to use the accent-conversion application discussed above in conjunction with an accent detection model, such that the computing device **300** is initially unaware of one or both accents that may be present in a given communication. For example, an accent detection model may be used in the initial moments of a conversation to identify the accents of the speakers. Based on the identified accents, the accent-conversion application may determine the appropriate learned linguistic representation(s) that should be used by the ASR engine **302** and the corresponding learned mapping between representations that should be used by the VC engine **304**. Additionally, or alternatively, the accent detection model may be used to provide a suggestion to a user for which input/output accent the user should select to obtain the best results. Other implementations incorporating an accent detection model are also possible.

FIG. **4** includes one or more operations, functions, or actions as illustrated by one or more of blocks **402-410**, respectively. Although the blocks are illustrated in sequential order, some of the blocks may also be performed in parallel, and/or in a different order than those described herein. Also, the various blocks may be combined into fewer blocks, divided into additional blocks, and/or removed based upon the desired implementation.

In addition, for the example flow chart in FIG. **4** and other processes and methods disclosed herein, the flow chart shows functionality and operation of one possible implementation of present embodiments. In this regard, each block may represent a module, a segment, or a portion of program code, which includes one or more instructions executable by one or more processors for implementing logical functions or blocks in the process.

The program code may be stored on any type of computer readable medium, for example, such as a storage device including a disk or hard drive. The computer readable medium may include non-transitory computer readable medium, for example, such as computer-readable media that stores data for short periods of time like register memory, processor cache and Random-Access Memory (RAM). The computer readable medium may also include non-transitory media, such as secondary or persistent long-term storage, like read only memory (ROM), optical or magnetic disks, compact disc read only memory (CD-ROM), for example. The computer readable media may also be any other volatile or non-volatile storage systems. The computer readable medium may be considered a computer readable storage medium, for example, or a tangible storage device. In addition, for the processes and methods disclosed herein, each block in FIG. **4** may represent circuitry and/or machinery that is wired or arranged to perform the specific functions in the process.

III. Conclusion

Example embodiments of the disclosed innovations have been described above. Those skilled in the art will understand, however, that changes and modifications may be made to the embodiments described without departing from the true scope and spirit of the present invention, which will be defined by the claims.

Further, to the extent that examples described herein involve operations performed or initiated by actors, such as "humans," "operators," "users," or other entities, this is for purposes of example and explanation only. Claims should not be construed as requiring action by such actors unless explicitly recited in claim language.

We claim:

1. A system, comprising:

at least one processor; and

non-transitory computer-readable medium comprising program instructions stored thereon that are executable by the at least one processor to cause the system to:

train a first machine-learning algorithm with first audio data comprising speech content captured from a plurality of different speakers having a first accent, wherein the training comprises aligning and classifying each of a first plurality of frames of the captured speech content corresponding to respective ones of the plurality of different speakers;

apply the first machine-learning algorithm to speech content received via at least one microphone, and comprising a set of phonemes associated with a first pronunciation of the received speech content, to derive a non-text linguistic representation of the set of phonemes associated with the first pronunciation of the received speech content;

based on the derived non-text linguistic representation of the set of phonemes associated with the first pronunciation of the received speech content, synthesize, using a second machine-learning-algorithm trained with (i) second audio data comprising the first accent and (ii) third audio data comprising a second accent, fourth audio data representative of the received speech content having the second accent, wherein synthesizing the fourth audio data comprises mapping at least a first non-text linguistic representation of a first phoneme of the set of phonemes associated with the first pronunciation of the received speech content to a second non-text linguistic representation of a second phoneme of an updated set of phonemes associated with a second pronunciation of the received speech content that is different from the first pronunciation of the received speech content, wherein the first and second phonemes are different phonemes; and

convert the synthesized fourth audio data into a synthesized version of the received speech content having the second accent, wherein the synthesized version of the received speech content having the second accent comprises the updated set of phonemes associated with the second pronunciation of the received speech content.

2. The system of claim **1**, wherein the program instructions are executable by the at least one processor to further cause the system to apply, to the derived non-text linguistic representation of the received speech content having the first accent, a learned mapping between the second audio data comprising the first accent and the third audio data comprising the second accent.

3. The system of claim **1**, wherein the program instructions are executable by the at least one processor to further cause the system to map each frame in the non-text linguistic representation of the set of phonemes associated with the first pronunciation of the received speech content to a corresponding frame in the non-text linguistic representation of the updated set of phonemes associated with the second pronunciation of the received speech content in order to map the non-text linguistic representation of the set of phonemes associated with the first pronunciation of the received speech content to the non-text linguistic representation of the

updated set of phonemes associated with the second pronunciation of the received speech content.

4. The system of claim **1**, wherein the second audio data comprising the first accent corresponds to a plurality of speakers having the first accent.

5. The system of claim **1**, wherein the third audio data comprising the second accent corresponds to a single speaker having the second accent.

6. The system of claim **1**, wherein the program instructions are executable by the at least one processor to further cause the system to receive a first user input indicating a selection of the first accent and a second user input indicating a selection of the second accent.

7. The system of claim **1**, wherein the first machine-learning algorithm comprises a non-text learned linguistic representation for the first accent and the program instructions are executable by the at least one processor to further cause the system to:

align and classify each of the first plurality of frames according to monophone and triphone sounds of the captured speech content to train the first machine-learning algorithm; and

determine, for each of a second plurality of frames in the received speech content, a respective (i) monophone and (ii) triphone sound detected in the frame based on the non-text learned linguistic representation for the first accent.

8. The system of claim **1**, wherein the program instructions are executable by the at least one processor to further cause the system to transmit the synthesized version of the received speech content having the second accent to a second computing device.

9. The system of claim **1**, wherein the program instructions are executable by the at least one processor to further cause the system to:

receive, in real time, continuous speech content having the first accent; and

continuously convert the synthesized fourth audio data into a synthesized version of the continuous speech content having the second accent between 50-700 ms after receiving the continuous speech content having the first accent.

10. The system of claim **1**, wherein the received speech content having the first accent further comprises a set of prosodic features, the program instructions are executable by the at least one processor to further cause the system to synthesize the fourth audio data representative of the received speech content having the second accent and the set of prosodic features, and the synthesized version of the received speech content having the second accent has the set of prosodic features.

11. A non-transitory computer-readable medium provisioned with program instructions that, when executed by at least one processor of a computing device, cause the computing device to:

train a first machine-learning algorithm with first audio data comprising speech content captured from a plurality of different speakers having a first accent, wherein the training comprises aligning and classifying each of a first plurality of frames of the captured speech content corresponding to respective ones of the plurality of different speakers;

apply the first machine-learning algorithm to speech content received via at least one microphone, and comprising a set of phonemes associated with a first pronunciation of the received speech content, to derive a non-text linguistic representation of the set of phonemes associated with the first pronunciation of the received speech content;

based on the derived non-text linguistic representation of the set of phonemes associated with the first pronunciation of the received speech content, synthesize, using a second machine-learning-algorithm trained with (i) second audio data comprising the first accent and (ii) third audio data comprising a second accent, fourth audio data representative of the received speech content having the second accent, wherein synthesizing the fourth audio data comprises mapping at least a first non-text linguistic representation of a first phoneme of the set of phonemes associated with the first pronunciation of the received speech content to a second non-text linguistic representation of a second phoneme of an updated set of phonemes associated with a second pronunciation of the received speech content that is different from the first pronunciation of the received speech content, wherein the first and second phonemes are different phonemes; and

convert the synthesized fourth audio data into a synthesized version of the received speech content having the second accent, wherein the synthesized version of the received speech content having the second accent comprises the updated set of phonemes associated with the second pronunciation of the received speech content.

12. The non-transitory computer-readable medium of claim **11**, wherein the program instructions, when executed by the at least one processor, further cause the computing device to apply, to the derived non-text linguistic representation of the received speech content having the first accent, a learned mapping between the second audio data comprising the first accent and the third audio data comprising the second accent.

13. The non-transitory computer-readable medium of claim **11**, wherein the program instructions, when executed by the at least one processor, further cause the computing device to map each frame in the non-text linguistic representation of the set of phonemes associated with the first pronunciation of the received speech content to a corresponding frame in the non-text linguistic representation of the updated set of phonemes associated with the second pronunciation of the received speech content in order to map the non-text linguistic representation of the set of phonemes associated with the first pronunciation of the received speech content to the non-text linguistic representation of the updated set of phonemes associated with the second pronunciation of the received speech content.

14. The non-transitory computer-readable medium of claim **11**, wherein one or more of the second audio data comprising the first accent corresponds to a plurality of speakers having the first accent or the third audio data comprising the second accent corresponds to a single speaker having the second accent.

15. The non-transitory computer-readable medium of claim **11**, wherein the program instructions, when executed by the at least one processor, further cause the computing device to receive a first user input indicating a selection of the first accent and a second user input indicating a selection of the second accent.

16. The non-transitory computer-readable medium of claim **11**, wherein the first machine-learning algorithm comprises a non-text learned linguistic representation for the first accent and the program instructions, when executed by the at least one processor, further cause the computing device to:

align and classify each of the first plurality of frames according to monophone and triphone sounds of the captured speech content to train the first machine-learning algorithm; and

determine, for each of a second plurality of frames in the received speech content, a respective (i) monophone and (ii) triphone sound detected in the frame based on the non-text learned linguistic representation for the first accent.

17. The non-transitory computer-readable medium of claim **11**, wherein the program instructions, when executed by the at least one processor, further cause the computing device to transmit the synthesized version of the received speech content having the second accent to a second computing device.

18. The non-transitory computer-readable medium of claim **11**, wherein the program instructions, when executed by the at least one processor, further cause the computing device to:

receive, in real time, continuous speech content having the first accent; and

continuously convert the synthesized fourth audio data into a synthesized version of the continuous speech content having the second accent between 50-700 ms after receiving the continuous speech content having the first accent.

19. A method comprising:

training a first machine-learning algorithm with first audio data comprising speech content captured from a plurality of different speakers having a first accent, wherein the training comprises aligning and classifying each of a first plurality of frames of the captured speech content corresponding to respective ones of the plurality of different speakers;

applying the first machine-learning algorithm to speech content received via at least one microphone, and comprising a set of phonemes associated with a first pronunciation of the received speech content, to derive a non-text linguistic representation of the set of phonemes associated with the first pronunciation of the received speech content;

based on the derived non-text linguistic representation of the set of phonemes associated with the first pronunciation of the received speech content, synthesizing, using a second machine-learning-algorithm trained with (i) second audio data comprising the first accent and (ii) third audio data comprising a second accent, fourth audio data representative of the received speech content having the second accent, wherein synthesizing the fourth audio data comprises mapping at least a first non-text linguistic representation of a first phoneme of the set of phonemes associated with the first pronunciation of the received speech content to a second non-text linguistic representation of a second phoneme of an updated set of phonemes associated with a second pronunciation of the received speech content that is different from the first pronunciation of the received speech content, wherein the first and second phonemes are different phonemes; and

converting the synthesized fourth audio data into a synthesized version of the received speech content having the second accent, wherein the synthesized version of the received speech content having the second accent comprises the updated set of phonemes associated with the second pronunciation of the received speech content.

20. The method of claim **19**, further comprising:

receiving, in real time, continuous speech content having the first accent; and

continuously converting the synthesized fourth audio data into a synthesized version of the continuous speech content having the second accent between 50-700 ms after receiving the continuous speech content having the first accent.

21. The method of claim **19**, further comprising mapping each frame in the non-text linguistic representation of the set of phonemes associated with the first pronunciation of the received speech content to a corresponding frame in the non-text linguistic representation of the updated set of phonemes associated with the second pronunciation of the received speech content in order to map the non-text linguistic representation of the set of phonemes associated with the first pronunciation of the received speech content to the non-text linguistic representation of the updated set of phonemes associated with the second pronunciation of the received speech content.

22. The method of claim **19**, wherein the first machine-learning algorithm comprises a non-text learned linguistic representation for the first accent and the method further comprises:

aligning and classifying each of the first plurality of frames according to monophone and triphone sounds of the captured speech content to train the first machine-learning algorithm; and

determining, for each of a second plurality of frames in the received speech content, a respective (i) monophone and (ii) triphone sound detected in the frame based on the non-text learned linguistic representation for the first accent.

* * * * *

EXHIBIT B

US012125496B1

(12) **United States Patent**
Zhang et al.

(10) **Patent No.: US 12,125,496 B1**
(45) **Date of Patent: Oct. 22, 2024**

(54) **METHODS FOR NEURAL NETWORK-BASED VOICE ENHANCEMENT AND SYSTEMS THEREOF**

(71) Applicant: **Sanas.ai Inc.**, Palo Alto, CA (US)

(72) Inventors: **Shawn Zhang**, Palo Alto, CA (US); **Lukas Pfeifenberger**, Salzburg (AT); **Jason Wu**, Santa Clara, CA (US); **Piotr Dura**, Warsaw (PL); **David Braude**, Edinburgh (GB); **Bajibabu Bollepalli**, Cottenham (GB); **Alvaro Escudero**, San Sebastian de los Reyes (ES); **Gokce Keskin**, Mountain View, CA (US); **Ankita Jha**, Bangalore (IN); **Maxim Serebryakov**, Palo Alto, CA (US)

(73) Assignee: **SANAS.AI INC.**, Palo Alto, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **18/644,959**

(22) Filed: **Apr. 24, 2024**

Related U.S. Application Data

(60) Provisional application No. 63/464,432, filed on May 5, 2023.

(51) **Int. Cl.**
G10L 15/00 (2013.01)
G10L 15/02 (2006.01)
(Continued)

(52) **U.S. Cl.**
CPC ***G10L 21/0232*** (2013.01); ***G10L 15/02*** (2013.01); ***G10L 15/063*** (2013.01);
(Continued)

(58) **Field of Classification Search**
CPC G10L 15/16; G10L 19/005; G10L 19/00; G10L 25/27; G10L 25/30; G10L 15/02;
(Continued)

(56) **References Cited**

U.S. PATENT DOCUMENTS

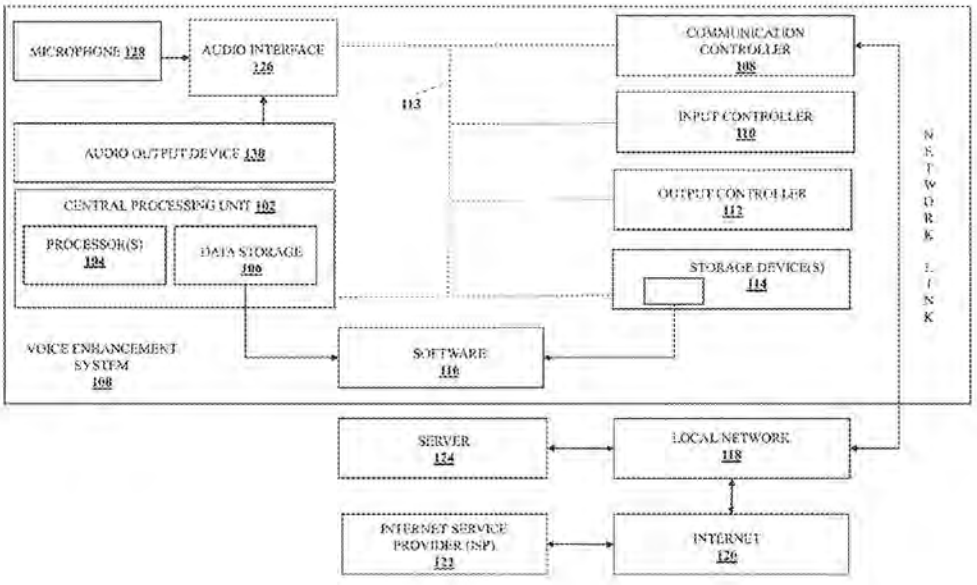
11,410,684 B1 * 8/2022 Klimkov G10L 25/78
11,482,235 B2 * 10/2022 Hsiung G06N 20/20
(Continued)

Primary Examiner — Vu B Hang
(74) *Attorney, Agent, or Firm* — Troutman Pepper Hamilton Sanders LLP

(57) **ABSTRACT**

The disclosed technology relates to methods, voice enhancement systems, and non-transitory computer readable media for real-time voice enhancement. In some examples, input audio data including foreground speech content, non-content elements, and speech characteristics is fragmented into input speech frames. The input speech frames are converted to low-dimensional representations of the input speech frames. One or more of the fragmentation or the conversion is based on an application of a first trained neural network to the input audio data. The low-dimensional representations of the input speech frames omit one or more of the non-content elements. A second trained neural network is applied to the low-dimensional representations of the input speech frames to generate target speech frames. The target speech frames are combined to generate output audio data. The output audio data further includes one or more portions of the foreground speech content and one or more of the speech characteristics.

20 Claims, 8 Drawing Sheets



(51) **Int. Cl.**

G10L 15/06	(2013.01)
G10L 21/0232	(2013.01)
G10L 25/30	(2013.01)
G10L 15/16	(2006.01)
G10L 15/22	(2006.01)

(52) **U.S. Cl.**
CPC ***G10L 25/30*** (2013.01); *G10L 15/16* (2013.01); *G10L 15/22* (2013.01)

(58) **Field of Classification Search**
CPC G10L 15/063; G10L 15/08; G10L 15/07; G10L 15/20; G10L 15/22; G10L 15/26; G10L 15/30; G10L 15/12; G10L 21/0208; G10L 25/78; G10L 25/87
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

11,705,147 B2 *	7/2023	Visser	.....	G06N 3/044 704/200
11,868,883 B1 *	1/2024	Commons	.....	G06F 40/30

* cited by examiner

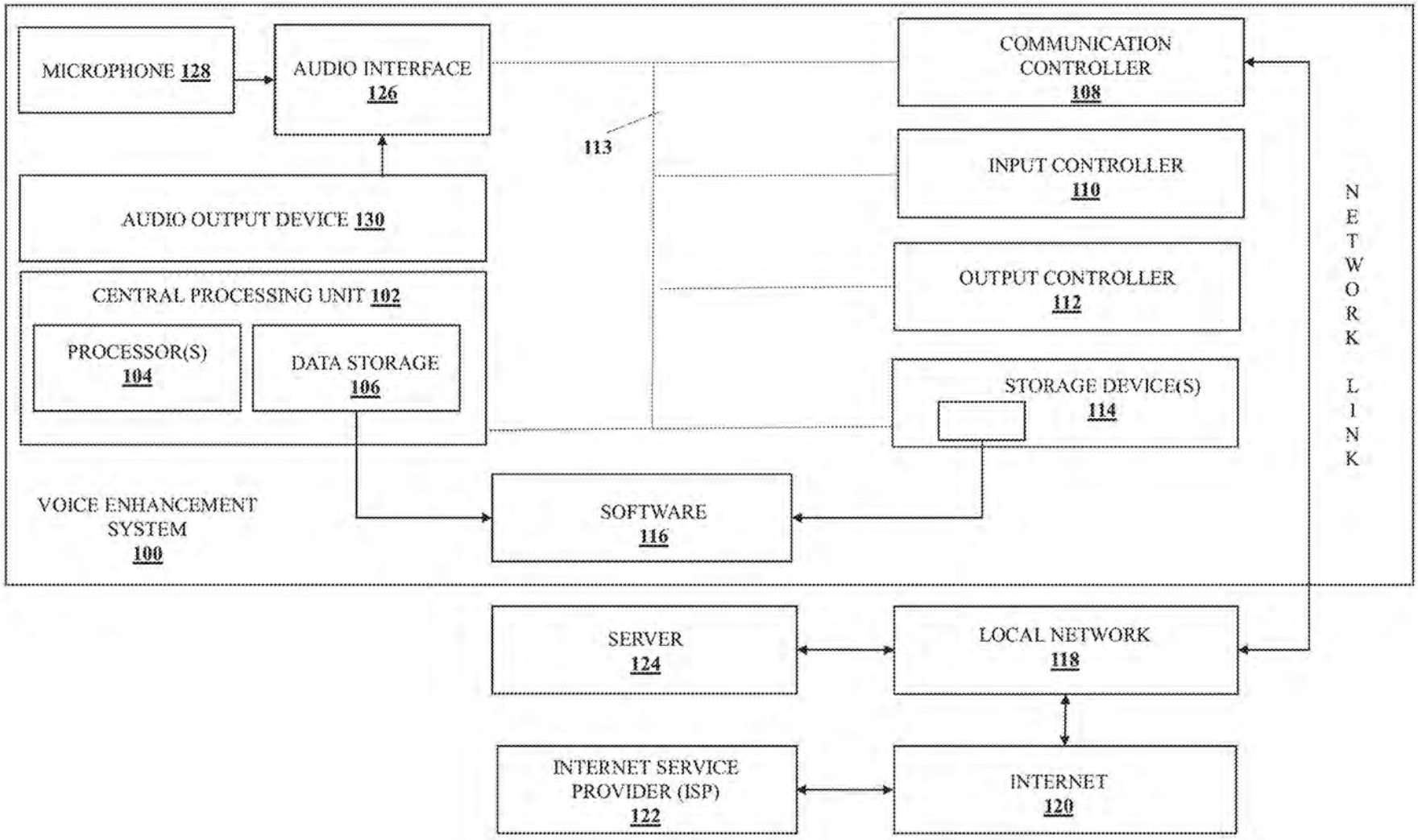

MICROPHONE 128
AUDIO INTERFACE 126
AUDIO OUTPUT DEVICE 130
CENTRAL PROCESSING UNIT 102
PROCESSOR(S) 104
DATA STORAGE 106
VOICE ENHANCEMENT SYSTEM 100
113
COMMUNICATION CONTROLLER 108
INPUT CONTROLLER 110
OUTPUT CONTROLLER 112
STORAGE DEVICE(S) 114
SOFTWARE 116
NETWORK LINK
SERVER 124
LOCAL NETWORK 118
INTERNET SERVICE PROVIDER (ISP) 122
INTERNET 120

FIG. 1

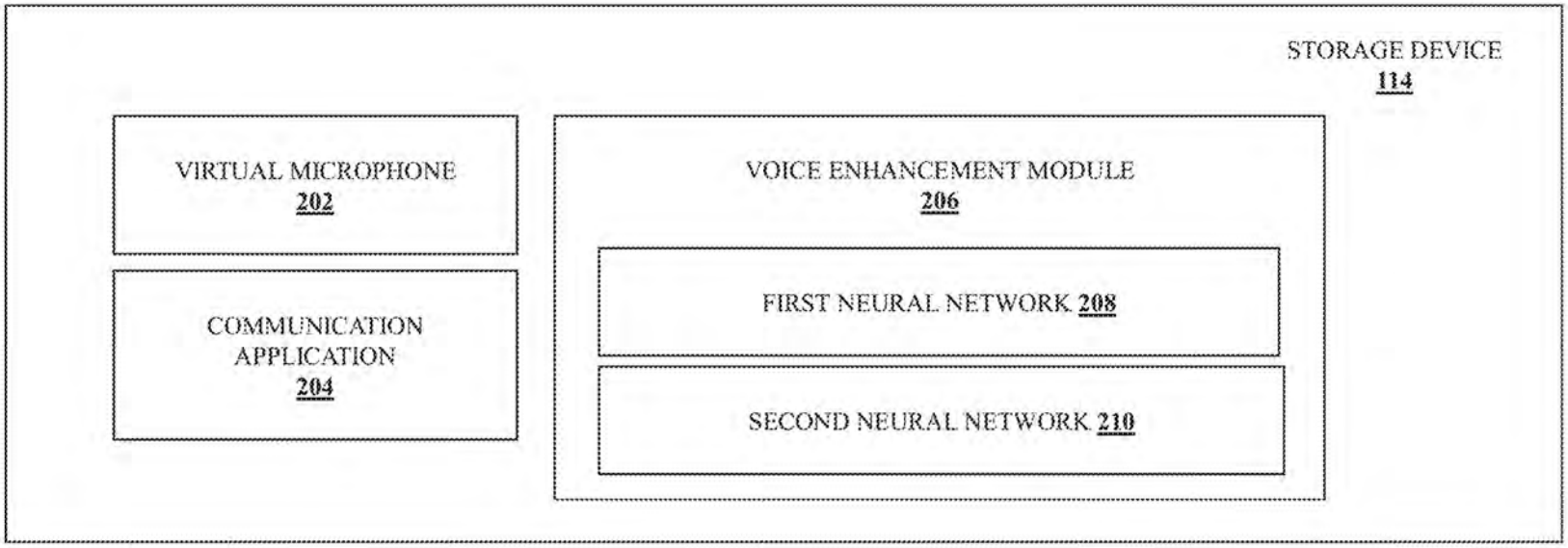

STORAGE DEVICE
114
VIRTUAL MICROPHONE
202
COMMUNICATION
APPLICATION
204
VOICE ENHANCEMENT MODULE
206
FIRST NEURAL NETWORK 208
SECOND NEURAL NETWORK 210

FIG. 2

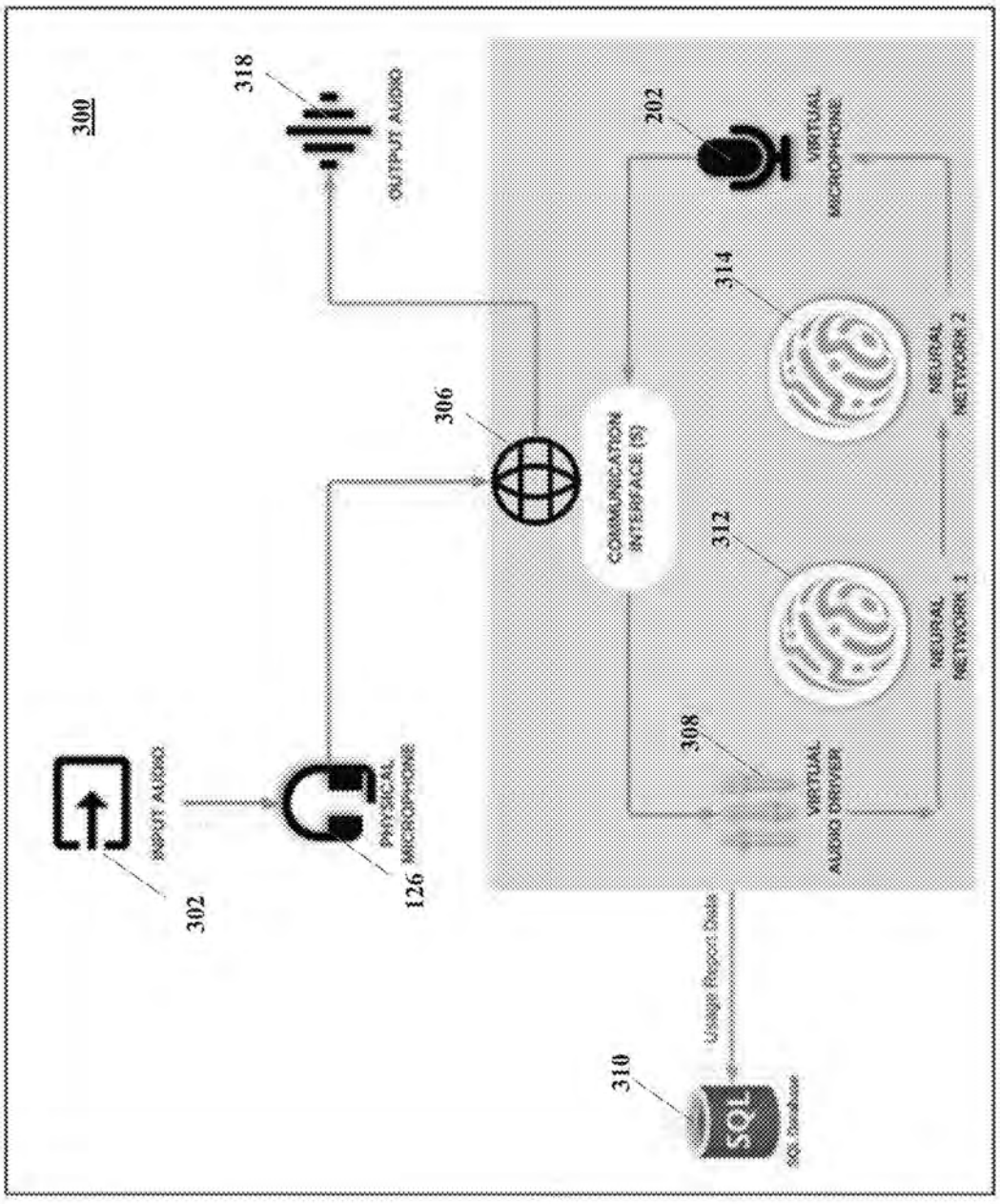

300
318
OUTPUT AUDIO
202
VIRTUAL MICROPHONE
314
NEURAL NETWORK 2
306
COMMUNICATION INTERFACE (S)
312
NEURAL NETWORK 1
308
VIRTUAL AUDIO DRIVER
INPUT AUDIO
302
PHYSICAL MICROPHONE
126
Usage Report Data
310
SQL

FIG. 3

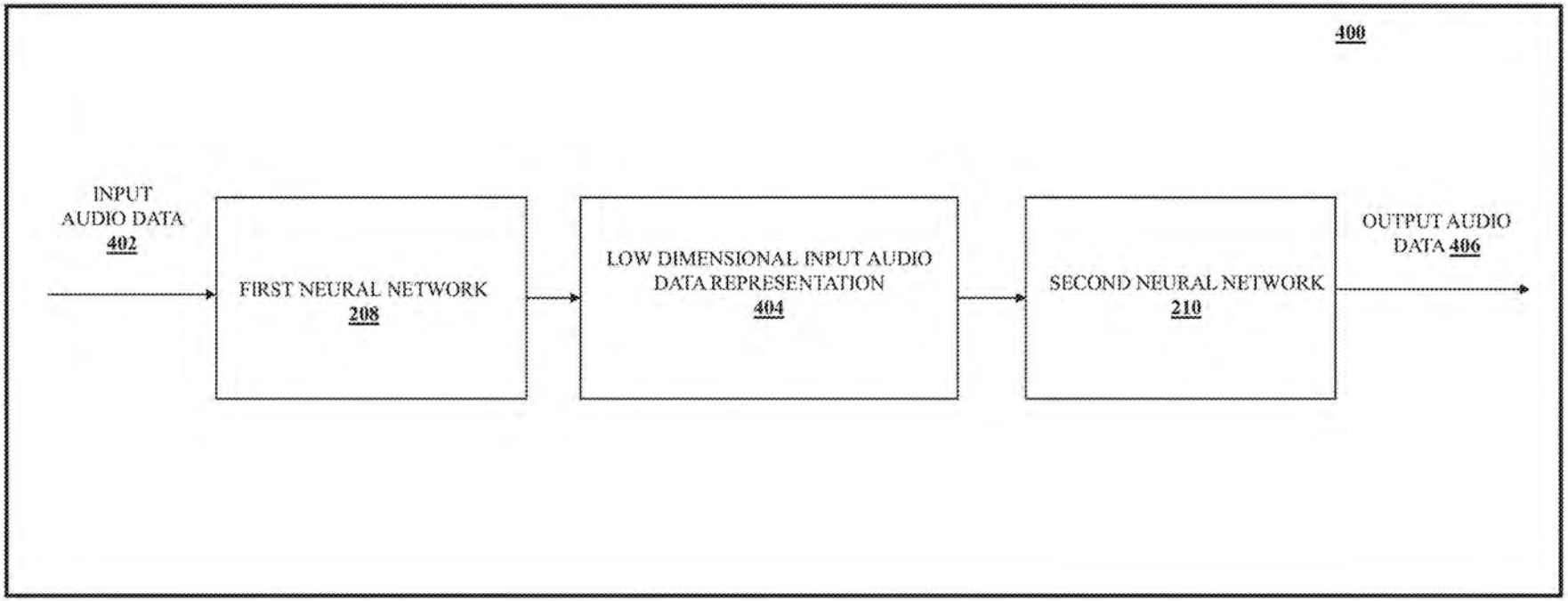

400
INPUT AUDIO DATA 402
FIRST NEURAL NETWORK 208
LOW DIMENSIONAL INPUT AUDIO DATA REPRESENTATION 404
SECOND NEURAL NETWORK 210
OUTPUT AUDIO DATA 406

FIG. 4

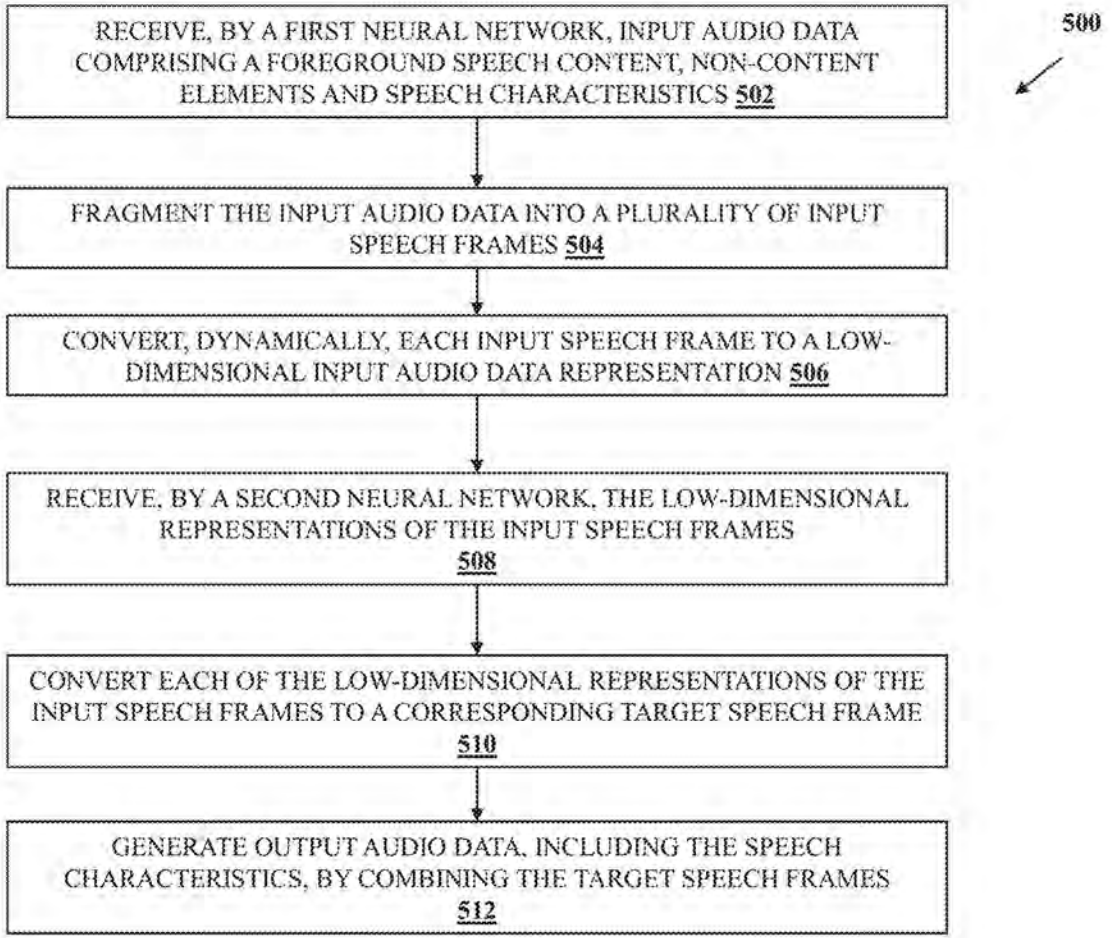

500
RECEIVE, BY A FIRST NEURAL NETWORK, INPUT AUDIO DATA COMPRISING A FOREGROUND SPEECH CONTENT, NON-CONTENT ELEMENTS AND SPEECH CHARACTERISTICS 502
FRAGMENT THE INPUT AUDIO DATA INTO A PLURALITY OF INPUT SPEECH FRAMES 504
CONVERT, DYNAMICALLY, EACH INPUT SPEECH FRAME TO A LOW-DIMENSIONAL INPUT AUDIO DATA REPRESENTATION 506
RECEIVE, BY A SECOND NEURAL NETWORK, THE LOW-DIMENSIONAL REPRESENTATIONS OF THE INPUT SPEECH FRAMES 508
CONVERT EACH OF THE LOW-DIMENSIONAL REPRESENTATIONS OF THE INPUT SPEECH FRAMES TO A CORRESPONDING TARGET SPEECH FRAME 510
GENERATE OUTPUT AUDIO DATA, INCLUDING THE SPEECH CHARACTERISTICS, BY COMBINING THE TARGET SPEECH FRAMES 512

FIG. 5

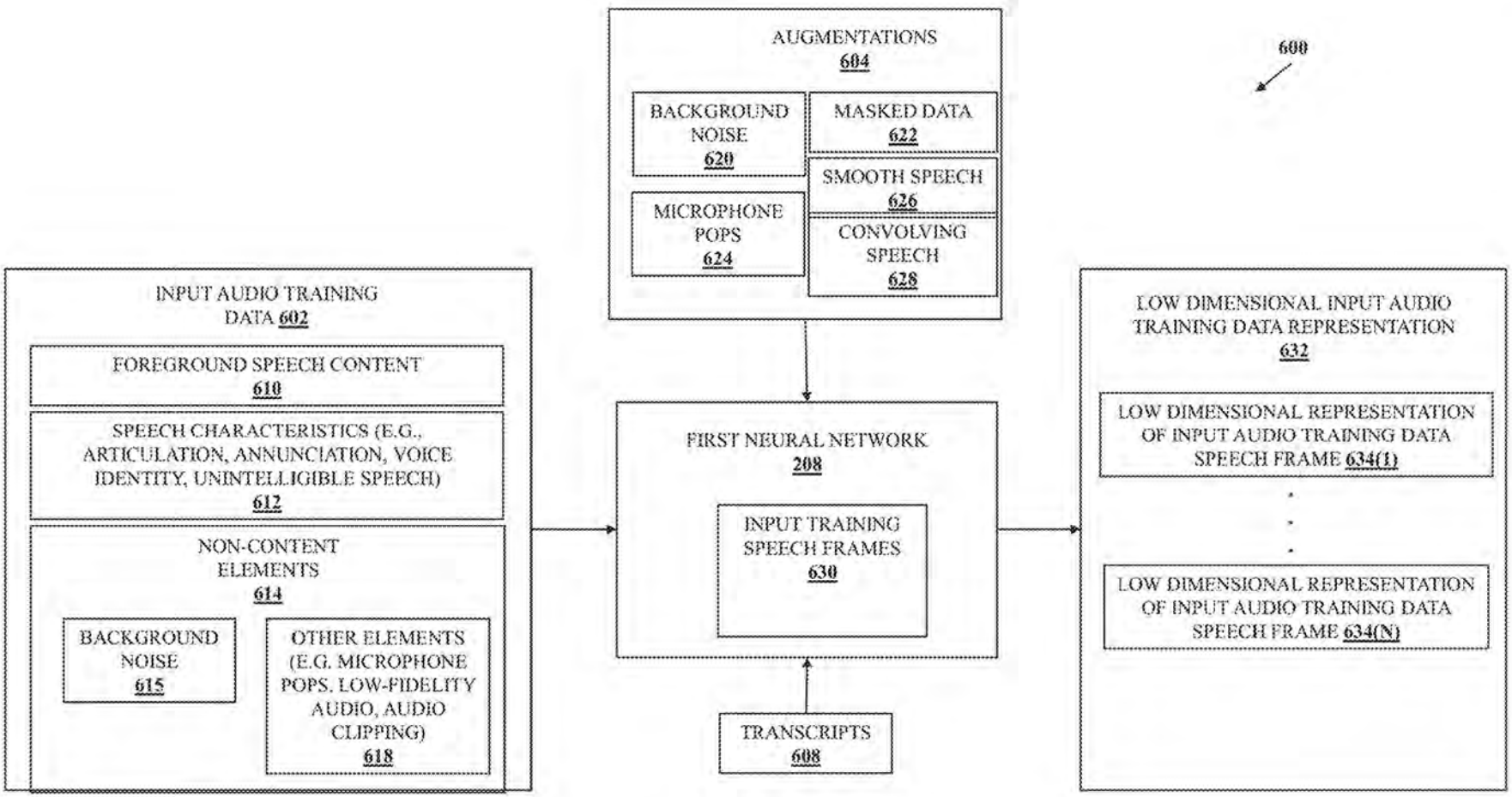

600
AUGMENTATIONS
604
BACKGROUND NOISE
620
MASKED DATA
622
SMOOTH SPEECH
626
MICROPHONE POPS
624
CONVOLVING SPEECH
628
INPUT AUDIO TRAINING DATA 602
FOREGROUND SPEECH CONTENT
610
SPEECH CHARACTERISTICS (E.G., ARTICULATION, ANNUNCIATION, VOICE IDENTITY, UNINTELLIGIBLE SPEECH)
612
NON-CONTENT ELEMENTS
614
BACKGROUND NOISE
615
OTHER ELEMENTS (E.G. MICROPHONE POPS, LOW-FIDELITY AUDIO, AUDIO CLIPPING)
618
FIRST NEURAL NETWORK
208
INPUT TRAINING SPEECH FRAMES
630
TRANSCRIPTS
608
LOW DIMENSIONAL INPUT AUDIO TRAINING DATA REPRESENTATION
632
LOW DIMENSIONAL REPRESENTATION OF INPUT AUDIO TRAINING DATA SPEECH FRAME 634(1)
LOW DIMENSIONAL REPRESENTATION OF INPUT AUDIO TRAINING DATA SPEECH FRAME 634(N)

FIG. 6

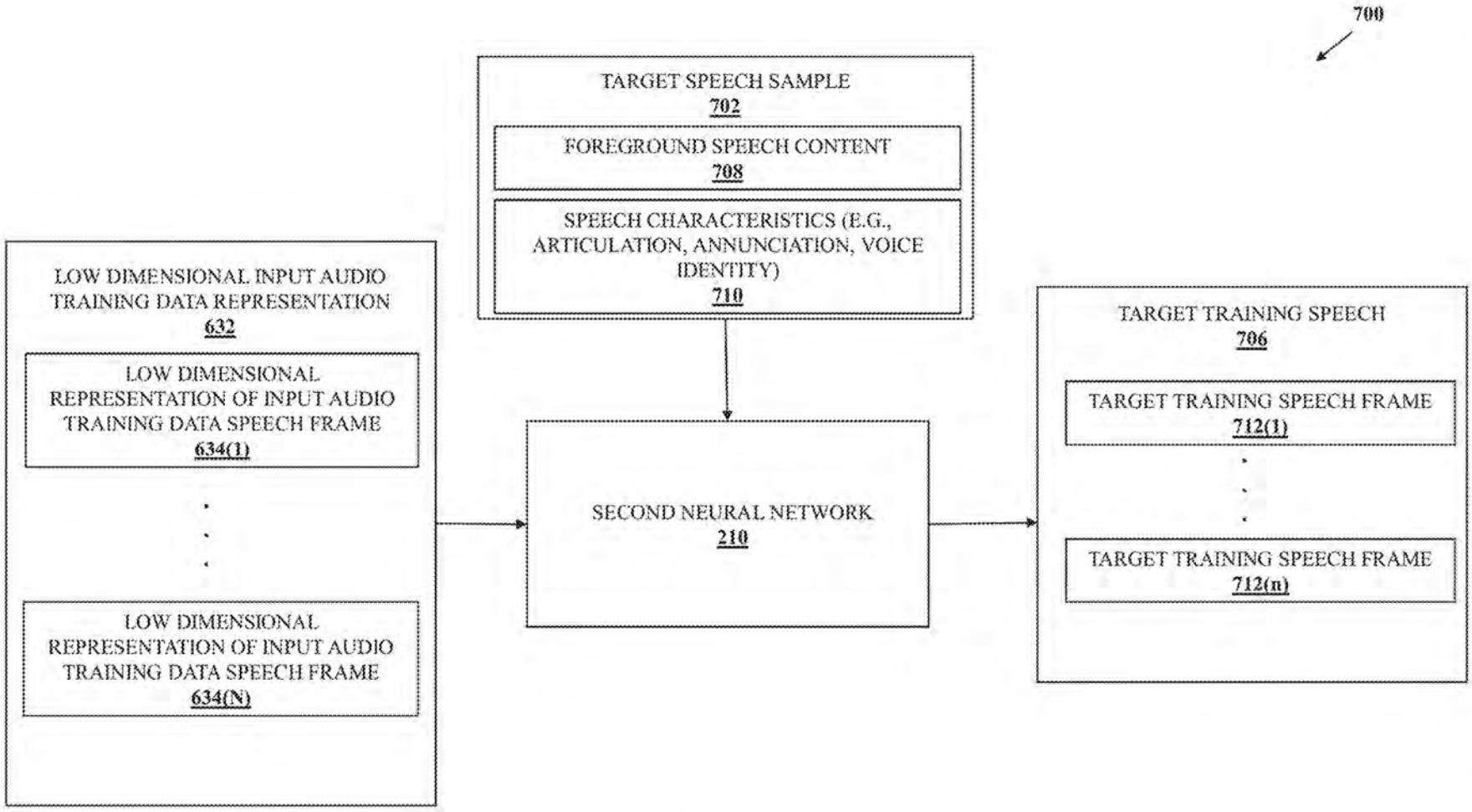

700
TARGET SPEECH SAMPLE
702
FOREGROUND SPEECH CONTENT
708
SPEECH CHARACTERISTICS (E.G., ARTICULATION, ANNUNCIATION, VOICE IDENTITY)
710
LOW DIMENSIONAL INPUT AUDIO TRAINING DATA REPRESENTATION
632
LOW DIMENSIONAL REPRESENTATION OF INPUT AUDIO TRAINING DATA SPEECH FRAME
634(1)
LOW DIMENSIONAL REPRESENTATION OF INPUT AUDIO TRAINING DATA SPEECH FRAME
634(N)
SECOND NEURAL NETWORK
210
TARGET TRAINING SPEECH
706
TARGET TRAINING SPEECH FRAME
712(1)
TARGET TRAINING SPEECH FRAME
712(n)

FIG. 7

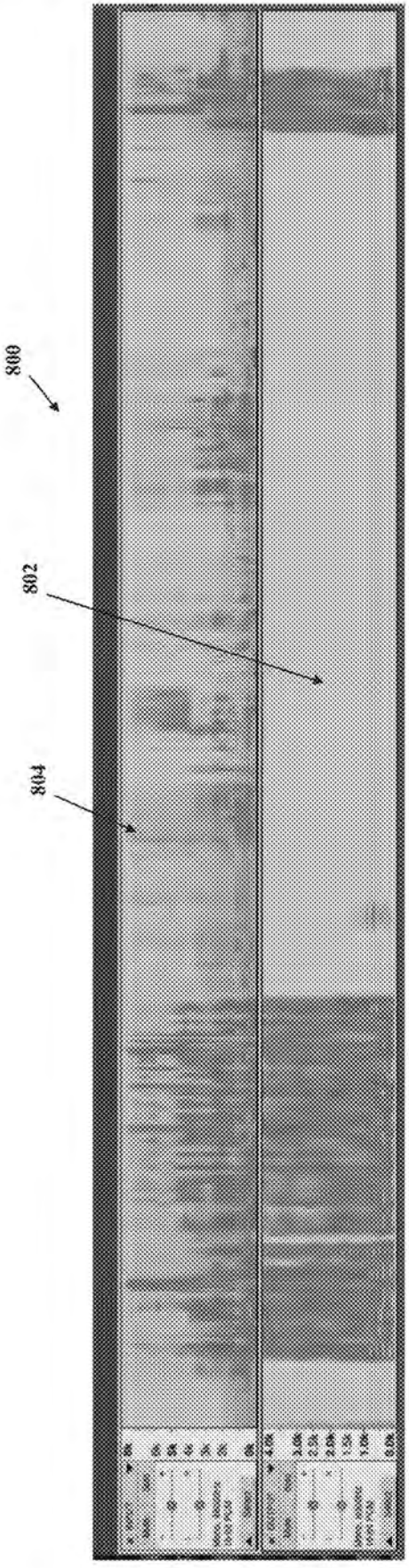


FIG. 8

METHODS FOR NEURAL NETWORK-BASED VOICE ENHANCEMENT AND SYSTEMS THEREOF

This application claims priority to U.S. Provisional Patent Application Ser. No. 63/464,432, filed May 5, 2023, which is hereby incorporated herein by reference in its entirety.

FIELD

This technology generally relates to audio analysis and, more particularly, to methods and systems for voice enhancement using neural networks.

BACKGROUND

Many environments, such as inside of a vehicle, a bustling street, or a busy office, are susceptible to disruptive noise that can obstruct speech. The level of background noise can range from the quiet humming of a computer fan to the noisy chatter of a crowded café. This noise can not only directly hinder a listener's ability to understand speech but also lead to further unwanted distortions when the speech is processed. Voice enhancement techniques can be employed to enhance quality and clarity of speech, often with a focus on reducing noise.

In customer service roles, for example, where clear communication is essential for customer satisfaction, voice enhancement is used to improve the quality of calls and reduce misunderstandings. In the medical field, voice enhancement technology is used to enhance the quality of recordings of medical consultations, which can be useful for training and research purposes. In education, voice enhancement technology is used to help students with hearing impairments understand lectures and discussions more clearly, and there are many other use cases and applications of voice enhancement technology.

One approach for voice enhancement and noise suppression in speech signals is through speech separation, which considers all background sounds as noise. Speech separation processing is often carried out in the short-time Fourier transform (STFT) domain. Ratio mask is another technique employed to distinguish speech signals from background noise, providing a means to diminish noise and enhance speech signals. Ratio mask leverages a representation of the signal-to-noise ratio (SNR) at each frequency band within an audio signal.

Another approach used in voice enhancement is equalization, which involves adjusting the frequency response of a speech signal to enhance its clarity and naturalness. The voice enhancement process involves regulating the level of various frequency components in the speech signal to improve the clarity of speech.

While current enhancement techniques can decrease noise and enhance the quality of the signal that is perceived, they can also distort the speech features that are necessary for speech recognition. This distortion caused by the suppression of noise can be more severe than the noise itself, which can result in inaccurate results when using automatic speech recognition (ASR) software. Additionally, current voice enhancement methods are only capable of attempting to preserve original speech audio, which can present a challenge when the original speech is unclear due to characteristics such as slurring, mumbling, or being too quiet.

For instance, a customer care representative may develop a sore throat and find it difficult to speak clearly on phone, while another representative may become fatigued and have

trouble speaking clearly after extended periods of speaking on the phone. Moreover, people with speech patterns that are naturally unclear or indistinct, such as mumbling, creakiness, slurring, or quiet speech, may find that these characteristics hinder their ability to speak clearly and be easily understood. In another example, people with speech disorders, such as dysarthria or apraxia, can make it difficult for them to communicate effectively.

Since many current voice enhancement methods focus on noise removal, they have reduced effectiveness when the speech itself is not intelligible. Other current voice enhancement techniques fail to sufficiently enhance the quality, clarity, comprehensibility, and intelligibility of degraded speech signals.

BRIEF DESCRIPTION OF THE DRAWINGS

The disclosed technology is illustrated by way of example and not limitation in the accompanying figures, in which like references indicate similar elements.

FIG. **1** is a block diagram of an exemplary network environment that includes a voice enhancement system.

FIG. **2** is a block diagram of an exemplary storage device of the voice enhancement system of FIG. **1**.

FIG. **3** is a flow diagram of a method for real-time voice enhancement.

FIG. **4** is a flow diagram of another method for real-time voice enhancement.

FIG. **5** is a flowchart of an exemplary method for real-time voice enhancement.

FIG. **6** is a schematic diagram of an exemplary method for training a first neural network.

FIG. **7** is a schematic diagram of another exemplary method for training a second neural network.

FIG. **8** is an exemplary representation of converting a low-dimensional representation of input speech frames to target speech frames.

DETAILED DESCRIPTION

Examples described below may be used to provide methods, devices (e.g., a non-transitory computer readable medium), apparatuses, and/or systems for neural network-based voice enhancement and noise suppression. Although the technology has been described with reference to specific examples, various modifications may be made to these examples without departing from the broader spirit and scope of the various embodiments of the technology described and illustrated by way of the examples herein. This technology advantageously improves speech clarity and intelligibility in various applications by utilizing noise suppression algorithms that more accurately estimate the background noise signal from a single microphone recording, thereby suppressing noise without distorting the target or output enhanced speech data.

Referring now to FIG. **1**, a block diagram of an exemplary network environment that includes a voice enhancement system **100** is illustrated. The voice enhancement system **100** in this example includes processor(s) **104**, which are designed to process instructions (e.g., computer readable instructions (i.e., code)) stored on the storage device(s) **114** (e.g., a non-transitory computer readable medium) of the voice enhancement system **100**. By processing the stored instructions, the processor(s) **104** may perform the steps and functions disclosed herein, such as with reference to FIG. **5**, for example.

The storage device(s) **114** may be optical storage device(s), magnetic storage device(s), solid-state storage device(s) (e.g., solid-state disks (SSDs)), non-transitory storage device(s), another type of memory, and/or a combination thereof, for example, although other types of storage device(s) can also be used. The storage device(s) **114** may contain software **116**, which is a set of instructions (i.e., program code). Alternatively, instructions may be stored in one or more remote storage devices, for example storage devices (e.g., hosted by a server **124**) accessed over a local network **118** or the Internet **120** via an Internet Service Provider (ISP) **122**.

The voice enhancement system **100** also includes an operating system and microinstruction code in some examples, one or both of which can be hosted by the storage device(s) **114**. The various processes and functions described herein may either be part of the microinstruction code and/or program code (or a combination thereof), which is executed via the operating system. The voice enhancement system **100** also may have data storage **106**, which along with the processor(s) **104** form a central processing unit (CPU) **102**, an input controller **110**, an output controller **112**, and/or a communication controller **108**. A bus **113** may operatively couple components of the voice enhancement system **100**, including processor(s) **104**, data storage **106**, storage device(s) **114**, input controller **110**, output controller **112**, and/or any other devices (e.g., a network controller or a sound controller).

The output controller **112** may be operatively coupled (e.g., via a wired or wireless connection) to a display device (e.g., a monitor, television, mobile device screen, touch-display, etc.) in such a fashion that output controller **112** can transform the display on the display device (e.g., in response to the execution of module(s)). Input controller **110** may be operatively coupled (e.g., via a wired or wireless connection) to an input device (e.g., mouse, keyboard, touchpad scroll-ball, touch-display, etc.) in such a fashion that input can be received from a user of the voice enhancement system **100**.

The communication controller **108** is coupled to a bus **113** in some examples and provides a two-way coupling through a network link to the Internet **120** that is connected to a local network **118** and operated by an ISP **122**, which provides data communication services to the Internet **120**. The network link typically provides data communication through one or more networks to other data devices. For example, a network link may provide a connection through local network **118** to a host computer and/or to data equipment operated by the ISP **122**. A server **124** may transmit requested code for an application through the Internet **120**, ISP **122**, local network **118**, and/or communication controller **108**.

The audio interface **126**, also referred to as a sound card, includes sound processing hardware and/or software, including a digital-to-analog converter (DAC) and an analog-to-digital converter (ADC). The audio interface **126** is coupled to a physical microphone **128** and an audio output device **130** (e.g., headphones or speaker(s)) in this example, although the audio interface **126** can be coupled to other types of audio devices in other examples. Thus, the audio interface **126** uses the ADC to digitize input analog audio signals from a sound source (e.g., the microphone **128**) so that the digitized signals can be processed by the voice enhancement system **100**, such as according to the methods described and illustrated herein. The DAC of the audio interface **126** can convert generated digital audio data into an analog format for output via the audio output device **130**.

The voice enhancement system **100** is illustrated in FIG. **1** with all components as separate devices for ease of identification only. One or more of the components of the voice enhancement system **100** in other examples may be separate devices (e.g., a personal computer connected by wires to a monitor and mouse), may be integrated in a single device (e.g., a mobile device with a touch-display, such as a smartphone or a tablet), or any combination of devices (e.g., a computing device operatively coupled to a touch-screen display device, a plurality of computing devices attached to a single display device and input device, etc.). The voice enhancement system **100** also may be one or more servers, for example a farm of networked or distributed servers, a clustered server environment, or a cloud network of computing devices. Other network topologies can also be used in other examples.

Referring now to FIG. **2**, a block diagram of an exemplary one of the storage device(s) **114** of the voice enhancement system **100** is illustrated. The storage device **114** may include a virtual microphone **202**, a communication application **204**, and a voice enhancement module **206** with a first neural network **208** and a second neural network **210**, although other types and/or number of applications or modules can also be included in the storage device **114** in other examples. The virtual microphone **202** receives input audio data (e.g., digitized input audio signals) from the physical microphone **128**, which is communicated to the voice enhancement module **206**.

The virtual microphone **202** then receives the output of the second neural network **210** from the voice enhancement module **206**, which represents output audio data including target speech that is an enhanced version of the input audio data and provides the output to the communication application **204**. The communication application **204** can be audio or video conferencing or other software that provides an interface to a user of the voice enhancement system **100**, for example.

Thus, the voice enhancement module **206** performs voice enhancement and/or noise suppression to convert the input audio data into the output audio data using the first and second neural networks **208** and **210**, respectively. The first neural network **208** receives input audio data, fragments the input audio data into frames, and converts the frames to low-dimensional representations, also referred to as a reduced-dimension representation, having lower dimensionality than that of the input audio data. The first neural network **208** can be trained as explained in more detail below with reference to FIG. **6**.

The second neural network **210** receives the low-dimensional representations of the frames, converts the low-dimensional representations to corresponding target speech frames, and generates target speech frames, and combines the target speech frames to generate output audio data. The second neural network **210** can be trained as explained in more detail below with reference to FIG. **7**. The operation of the voice enhancement module **206** is described in more detail below with reference to FIG. **5**. In some examples, the virtual microphone **202** and the voice enhancement module **206** are combined within the same software application or other type of module.

Referring now to FIG. **3**, a flow diagram of a method **300** for real-time voice enhancement is illustrated. In this example, a user of the voice enhancement system **100** may provide input audio **302** via analog audio signals received by a physical microphone **128** of the voice enhancement system **100** and subsequently digitized by the audio interface **126**. The physical microphone **128** can be an integrated compo-

nent of the voice enhancement system **100** (e.g., an onboard microphone of a laptop computer or smartphone). In other examples, the physical microphone **128** can be a wired or wireless peripheral device (e.g., a webcam or a dedicated hardware microphone) that is connected to an I/O interface of the voice enhancement system **100**, and other exemplary physical microphones can also be used in yet other examples.

The digitized input audio **302** in this example is then routed from the physical microphone **128** over a communication interface **306** to a virtual audio driver **308**. Advantageously, the voice enhancement may be accomplished locally on the voice enhancement system **100** in examples in which the communication interface **306** is the bus **113**, which may minimize latency as compared to deployments that utilize cloud-based computing in which the communication interface **306** is the local network **118** and/or the Internet **120**, for example. Optionally, usage report data can be generated and maintained in a local or remote database **310**.

The digitized input audio **302** is then routed from the virtual audio driver **308** to a first neural network **208** and a second neural network **210** to enhance the voice and/or suppress the noise in the input audio **302**, as described and illustrated in more detail below. The output of the second neural network **210** is a digital version of the input audio **302** converted according to the voice enhancement and/or noise suppression methods described and illustrated herein, which is provided to a virtual microphone **202** executed by the voice enhancement system **100**. The virtual microphone **202** in this example uses the communication interface **306** to provide analog output audio **318** corresponding to the converted input audio **302**.

Accordingly, in some examples, the software **116** that facilitates the voice enhancement and/or noise suppression may function as the virtual microphone **202** that receives the input audio **302** from the physical microphone **128** and performs voice enhancement and/or noise suppression to convert the input audio **302** into the output audio **318**, as explained herein. The virtual microphone **202** then routes the converted output audio **318** via the communication interface **306** to the communication application **204** (e.g., Zoom™, Skype™, Viber™, Telegram™, etc.) executed by the voice enhancement system **100**, which would otherwise receive the input audio **302** directly from the physical microphone **128** without the technology described and illustrated by way of the examples herein.

Referring now to FIG. **4**, a flow diagram of another method **400** for real-time voice enhancement is illustrated. In this example, the voice enhancement system **100** applies the first neural network **208** to received input audio **302** that has been digitized to generated input audio data **402**. The first neural network dynamically converts the input audio data **402** to a low-dimensional input audio data representation **404**.

The voice enhancement system **100** then applies the second neural network **210** to the low-dimensional input audio data representation **404** to dynamically generate output audio data **406**, which can be converted to analog signals before being output as output audio **318**. The target speech of the output audio data **406** has enhanced voice and/or suppressed noise as compared to the input speech of the input audio data **402** as a result of the application of the first and second neural networks **208** and **210**, respectively. The output audio data **406** can then be output or provided, such as to the digital communication application **204**, for example, as explained above.

Referring to FIG. **5**, a flowchart of an exemplary method **500** for real-time voice enhancement is illustrated. In step **502** in this particular example, the voice enhancement system **100** provides to the first neural network **208** input audio data **402** including foreground speech content, one or more non-content elements, and a set of speech characteristics.

Referring to FIG. **6**, a schematic diagram of an exemplary method **600** for training the first neural network **208** is illustrated. In this example, the first neural network **208** may be trained with input audio training data **602**, one or more augmentations **604**, and one or more transcripts **608**, although additional training data can also be used in other examples. The input audio training data **602** in this example includes foreground speech content **610**, a set of speech characteristics **612**, and one or more non-content elements **614**.

The speech characteristics **612** may include one or more of pitch, intonation, melody, stress, articulation, annunciation, voice identity, and/or unintelligible speech, for example. The unintelligible speech can be caused by one or more factors such as background noise, poor enunciation, heavy accents, language barriers and/or mumbled, creaky, slurred, and/or quiet speech, for example.

In some examples, the non-content elements **614** may include background noise **616** and other elements **618** such as microphone pops, low-fidelity audio, and/or audio clipping, although other types of background noise can also be used. The augmentations **604** may include background noise **620**, masked data **622**, microphone pops **624**, smooth speech **626**, and/or convolving speech **628**, although other augmentations can also be used in other examples. The augmentations in this example are included to simulate degraded speech characteristics.

The input audio training data **602** in this example may be fragmented into a plurality of input training speech frames **630**. Input training speech frames **630** may be converted dynamically to a low-dimensional input audio training data representation **632** by the first neural network **208**. The low-dimensional input audio training data representation **632** may comprise multiple low-dimensional representations of input audio training data speech frames **634(1)**-**634(*n*)**. The low-dimensional input audio training data representation **632** may further include one or more portions of the foreground speech content **610** and/or the speech characteristics **612**. Other methods for training the first neural network **208** can also be used in other examples.

Thus, the first neural network **208** may be optimized by the voice enhancement system **100** to learn a mapping between the input training speech frames and the low-dimensional input audio data training data representation **632**, using techniques such as supervised learning or reinforcement learning, for example. The first neural network **208** also may be fine-tuned by the voice enhancement system **100** using additional data to improve the performance, and the hyperparameters of the first neural network **208** may be optimized to obtain improved results.

Referring back to FIG. **5**, in step **504**, the voice enhancement system **100** applying the first neural network **208** fragments the input audio data **402** received in step **502** into a plurality of input speech frames. In step **506**, the voice enhancement system **100** applying the first neural network **208** dynamically converts each of the input speech frames fragmented in step **504** to a low-dimensional input audio data representation **404**.

In some examples, the low-dimensional input audio data representation **404** comprises foreground speech content and

at least one or more of the speech characteristics of the audio data received in step **502**. The low-dimensional input audio data representation **404** may omit any number of the non-content elements of the audio data received in step **502** (e.g., background noise, and other elements such as microphone pops, low-fidelity audio, and audio clippings).

In other examples, the low-dimensional input audio data representation **404** generated by the first neural network **208** may be achieved by pre-processing the input audio data **402** to remove noise and other distortions that may affect the quality of the speech signal. For example, a noise reduction algorithm may be applied to remove background noise, or a filtering technique may be used to remove high-frequency noise or pops.

Once the input audio data **402** is optionally pre-processed, features may be extracted by the voice enhancement system **100** such as by using Fourier Transform, Mel-Frequency Cepstral Coefficients (MFCC), or other techniques. These extracted features capture important characteristics of the resulting speech signal such as pitch, intonation, and formants, for example. The extracted features may be encoded by the voice enhancement system **100** into the low-dimensional input audio data representation **404** in step **506** using techniques such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), or other dimensionality reduction techniques, for example. The resulting low-dimensional input audio data representation **404** may capture the most important characteristics of the resulting speech signal while reducing the computational complexity of the first neural network **208**.

In some examples, the low-dimensional input audio data representation **404** of the input speech may be achieved by using a hierarchical feature extraction network that extracts multiple levels of features from the input audio data **402**. Each level of the network could be designed to capture different aspects of the input audio data **402**, such as frequency content, temporal dynamics, and/or speech characteristics, for example. At each level of the hierarchical feature extraction network, the extracted features could be compressed into a low-dimensional input audio data representation **404** using a compression algorithm such as principal component analysis (PCA) or non-negative matrix factorization (NMF), for example.

The resulting compressed features may be passed to the next level of the hierarchical feature extraction network for further processing. This approach advantageously captures more detailed aspects of the input audio data **402** than traditional methods that rely on a single, fixed feature representation. The use of compression algorithms allows for efficient processing and storage of the feature representations, which may improve the accuracy and efficiency of real-time voice enhancement by providing a more detailed and robust representation of the input audio data **402**.

In step **508**, the voice enhancement system **100** provides to the second neural network **210** the low-dimensional input audio data representation **404** generated in step **508**. Referring now to FIG. **7**, a schematic diagram of an exemplary method **700** for training the second neural network **210** is illustrated. The second neural network **210** may be trained using target speech sample **702** and low-dimensional representation of input training speech **704** to dynamically generate target training speech **706**. The target speech sample **702** may include foreground speech content **708** and/or speech characteristics **710** (e.g., articulation, annunciation, voice identity, and/or unintelligible speech). The foreground speech content **708** and/or the speech characteristics **710** may be the same or different than the foreground speech content **610** and the speech characteristics **612**, respectively.

The second neural network **210** may receive the low-dimensional input audio training data representation **632** and convert each of the low-dimensional representation of input audio training data speech frames **634(1)-634(*n*)** to a respective corresponding one of the target training speech frames **712(1)-712(*n*)**. The target training speech **706** can include one or more of the speech characteristics **710** and can be generated dynamically by combining the target training speech frames **712(1)-712(*n*)**. Other methods for training the second neural network **210** can also be used in other examples.

In some examples, the second neural network **210** is trained to convert each of the low-dimensional representation of input audio training data speech frames **634(1)-634(*n*)** with the respective corresponding one of the target training speech frames **712(1)-712(*n*)** in real-time, which may be achieved using dynamic conversion. Dynamic conversion may allow for the efficient processing of the input audio data **402**, ensure that the resulting target speech of the output audio data **406** may contain the desired speech characteristics, and enable real-time voice enhancement without the need for a separate conversion step.

Thus, the second neural network **210** may be initially trained using supervised learning to convert the low-dimensional representation of input audio training data speech frames **634(1)-634(*n*)** in real-time. The second neural network **210** may be trained to learn the conversion between the low-dimensional representation of input audio training data speech frames **634(1)-634(*n*)** and the target training speech frames **712(1)-712(*n*)** using a loss function that minimizes the difference between the predicted and actual target speech frames, for example.

Once the second neural network **210** is trained using supervised learning, it may be further fine-tuned using an unsupervised learning approach. The second neural network **210** may be trained to learn the underlying structure of the low-dimensional representation of input audio training data speech frames **634(1)-634(*n*)** without being provided with explicit target training speech frames, which may be achieved by training the second neural network **210** to predict future speech frames from past speech frames, without any knowledge of the target training speech frames. This training approach may help the second neural network **210** learn more robust and generalizable low-dimensional representation of input audio training data speech frames, which may be useful for converting input speech frames in real-time.

In yet other examples, diffusion probabilistic model(s), flow-based model(s), and/or generative adversarial network (GAN)-based model(s) can be used for the second neural network **210**. Using diffusion probabilistic models, the second neural network **210** can be trained to iteratively refine relatively noisy input audio data **402** to generate relatively high-quality speech in the output audio data **406**. Flow-based models are configured to learn transformations to map the distribution of relatively noisy input audio data **402** to relatively high-quality speech in the output audio data **406**. Additionally, GAN-based models can be used to train a "discriminator" for the second neural network **210** to distinguish between relatively poor-quality speech in the input audio data **402** and relatively high-quality speech in the output audio data **406**. Other types of models can also be used to train the second neural network **210** in other examples.

Referring back to FIG. **5**, in step **510**, the voice enhancement system **100** applying the second neural network **210** converts each frame of the low-dimensional input audio data representation **404** to a corresponding target speech frame (e.g., a frame of output audio data **406**). In some examples, converting each frame of the low-dimensional input audio data representation **404** to a corresponding target speech frame may involve using unsupervised learning algorithms, such as clustering or dimensionality reduction techniques, to identify patterns and relationships within the frames of the low-dimensional input audio data representation **404** and target speech frames.

In other examples, converting each frame of the low-dimensional input audio data representation **404** to a corresponding target speech frame may involve using reinforcement learning algorithms to train the second neural network **210** to optimize the conversion process by adjusting a set of parameters in real-time based on feedback from the generated output audio data **406**. This may allow the conversion process to adapt and improve over time based on the specific characteristics of the input speech and the desired speech characteristics.

In step **512**, the voice enhancement system **100** applying the second neural network **210** combines the target speech frames to dynamically generate the output audio data **406** that includes the target speech and one or more of the speech characteristics received in step **502**. The patterns learned in step **510** may be used in step **512** to generate the enhanced speech signal, which is also referred to herein as the output audio data **406**.

Referring to FIG. **8**, an exemplary representation **800** of converting a low-dimensional representation **404** of input speech frames to target speech frames is illustrated. The output or target speech **802** shown in the representation **800** and generated based on the technology described and illustrated herein, may advantageously preserve the speech characteristics and enhance the quality, clarity, comprehensibility, and/or intelligibility of degraded speech signals of the input speech **804**.

Having thus described the basic concept of the invention, it will be rather apparent to those skilled in the art that the foregoing detailed disclosure is intended to be presented by way of example only and is not limiting. Various alterations, improvements, and modifications will occur and are intended for those skilled in the art, though not expressly stated herein. These alterations, improvements, and modifications are intended to be suggested hereby, and are within the spirit and scope of the invention. Additionally, the recited order of processing elements or sequences, or the use of numbers, letters, or other designations, therefore, is not intended to limit the claimed processes to any order except as may be specified in the claims. Accordingly, the invention is limited only by the following claims and equivalents thereto.

What is claimed is:

1. A voice enhancement system, comprising memory having instructions stored thereon and one or more processors coupled to the memory and configured to execute the instructions to:

fragment input audio data into a plurality of input speech frames, wherein the input audio data comprises foreground speech content, one or more non-content elements, and one or more speech characteristics;

convert the input speech frames to low-dimensional representations of the input speech frames, wherein one or more of the fragmentation or the conversion is based on an application of a first neural network to the input audio data and the low-dimensional representations of the input speech frames omit one or more of the non-content elements;

apply a second neural network to the low-dimensional representations of the input speech frames to generate target speech frames; and

combine the target speech frames to generate output audio data, wherein the output audio data further comprises one or more portions of the foreground speech content and one or more of the speech characteristics.

2. The voice enhancement system of claim **1**, further comprising a physical microphone and an audio output device, wherein the one or more processors are further configured to execute the instructions to:

digitize analog input audio signals obtained via the physical microphone to generate the input audio data;

convert the output audio data to analog audio output signals; and

provide the analog audio output signals to the audio output device via one or more of a virtual microphone or a communication application executed by the voice enhancement system.

3. The voice enhancement system of claim **1**, wherein the non-content elements comprise one or more of background noise, microphone pops, low-fidelity audio, or audio clippings and the speech characteristics comprise one or more of pitch, intonation, melody, stress, articulation, annunciation, voice identity, or unintelligible speech.

4. The voice enhancement system of claim **1**, wherein the one or more processors are further configured to execute the instructions to train the first neural network using input audio training data, one or more augmentations, and one or more transcripts, wherein the first neural network is trained to learn a mapping between input training speech frames fragmented from the input audio training data and low-dimensional representation of input audio training data speech frames.

5. The voice enhancement system of claim **4**, wherein the augmentations simulate one or more degraded speech characteristics and comprise one or more of background noise, masked data, microphone pops, smooth speech, or convolving speech.

6. The voice enhancement system of claim **4**, wherein the one or more processors are further configured to execute the instructions to train the second neural network using a target speech sample and the low-dimensional representation of input audio training data speech frames, wherein the second neural network is trained to use dynamic conversion to learn a mapping between each of the low-dimensional representation of input audio training data speech frames and a corresponding one of a plurality of target training speech frames.

7. The voice enhancement system of claim **1**, wherein the second neural network comprises one or more of a diffusion probabilistic model, a flow-based model, or a generative adversarial network-based model.

8. The voice enhancement system of claim **1**, wherein the one or more processors are further configured to execute the instructions to pre-process the input audio data by applying one or more of a noise reduction algorithm to remove background noise from the input audio data or a filtering technique to remove high-frequency noise or pops from the input audio data.

9. The voice enhancement system of claim **1**, wherein the one or more processors are further configured to execute the instructions to:

extract one or more features from the input audio data, wherein the features comprise one or more of pitch, intonation, or formants; and

encode the extracted features into one or more of the low-dimensional representations of the input speech frames using a dimensionality reduction technique.

10. The voice enhancement system of claim **9**, wherein the one or more processors are further configured to execute the instructions to extract the features using a hierarchical feature extraction network comprises a plurality of levels, wherein each of the levels is configured to capture a different one or more of the features and the captured different one or more of the features are compressed at each of the levels.

11. A method for real-time voice enhancement, the method implemented by a voice enhancement system and comprising:

training a first neural network using input audio training data, one or more augmentations, and one or more transcripts and a second neural network using a target speech sample and a plurality of low-dimensional representation of input audio training data speech frames,

applying the trained first neural network to convert input speech frames fragmented from input audio data to low-dimensional representations of the input speech frames, wherein the low-dimensional representations of the input speech frames omit one or more non-content elements of the input audio data;

applying the trained second neural network to the low-dimensional representations of the input speech frames to generate target speech frames; and

combining the target speech frames to generate output audio data, wherein the output audio data further comprises one or more portions of foreground speech content of the input audio data and one or more speech characteristics of the input audio data.

12. The method of claim **11**, wherein the trained first neural network is trained to learn a mapping between input training speech frames fragmented from the input audio training data and the low-dimensional representation of input audio training data speech frames.

13. The method of claim **11**, wherein the augmentations simulate one or more degraded speech characteristics and comprise one or more of background noise, masked data, microphone pops, smooth speech, or convolving speech.

14. The method of claim **11**, further comprising preprocessing the input audio data by applying one or more of a noise reduction algorithm to remove background noise from the input audio data or a filtering technique to remove high-frequency noise or pops from the input audio data.

15. The method of claim **11**, further comprising:

extracting one or more features from the input audio data, wherein the features comprise one or more of pitch, intonation, or formants; and

encoding the extracted features into one or more of the low-dimensional representations of the input speech frames using a dimensionality reduction technique.

16. A non-transitory computer-readable medium comprising instructions that, when executed by at least one processor, cause the at least one processor to:

digitize analog input audio signals to generate input audio data;

fragment the input audio data into a plurality of input speech frames, wherein the input audio data comprises foreground speech content, one or more non-content elements, and one or more speech characteristics;

convert the input speech frames to low-dimensional representations of the input speech frames, wherein one or more of the fragmentation or the conversion is based on an application of a first neural network to the input audio data and the low-dimensional representations of the input speech frames omit one or more of the non-content elements;

apply a second neural network to the low-dimensional representations of the input speech frames to generate target speech frames;

combine the target speech frames to generate output audio data, wherein the output audio data further comprises one or more portions of the foreground speech content and one or more of the speech characteristics; and

convert the output audio data to analog audio output signals before providing the analog audio output signals to an audio output device.

17. The non-transitory computer-readable medium of claim **16**, wherein the non-content elements comprise one or more of background noise, microphone pops, low-fidelity audio, or audio clippings and the speech characteristics comprise one or more of pitch, intonation, melody, stress, articulation, annunciation, voice identity, or unintelligible speech.

18. The non-transitory computer-readable medium of claim **16**, wherein the instructions, when executed by the at least one processor further causes the at least one processor to train the first neural network using input audio training data, one or more augmentations, and one or more transcripts, wherein the first neural network is trained to learn a mapping between input training speech frames fragmented from the input audio training data and low-dimensional representation of input audio training data speech frames.

19. The non-transitory computer-readable medium of claim **18**, wherein the instructions, when executed by the at least one processor further causes the at least one processor to train the second neural network using a target speech sample and the low-dimensional representation of input audio training data speech frames, wherein the second neural network is trained to use dynamic conversion to learn a mapping between each of the low-dimensional representation of input audio training data speech frames and a corresponding one of a plurality of target training speech frames.

20. The non-transitory computer-readable medium of claim **16**, wherein the second neural network comprises one or more of a diffusion probabilistic model, a flow-based model, or a generative adversarial network-based model.

* * * * *

EXHIBIT C

US012131745B1

(12) **United States Patent**
Pfeifenberger et al.

(10) **Patent No.: US 12,131,745 B1**
(45) **Date of Patent: Oct. 29, 2024**

(54) **SYSTEM AND METHOD FOR AUTOMATIC ALIGNMENT OF PHONETIC CONTENT FOR REAL-TIME ACCENT CONVERSION**

(71) Applicant: **Sanas.ai Inc.**, Palo Alto, CA (US)

(72) Inventors: **Lukas Pfeifenberger**, Salzburg (AT); **Shawn Zhang**, Palo Alto, CA (US)

(73) Assignee: **SANAS.AI INC.**, Palo Alto, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **18/754,280**

(22) Filed: **Jun. 26, 2024**

Related U.S. Application Data

(60) Provisional application No. 63/510,487, filed on Jun. 27, 2023.

(51) **Int. Cl.**

G10L 21/007	(2013.01)
G06F 3/16	(2006.01)
G10L 13/00	(2006.01)
G10L 13/033	(2013.01)
G10L 15/02	(2006.01)

(Continued)

(52) **U.S. Cl.**

CPC ***G10L 21/007*** (2013.01); ***G06F 3/162*** (2013.01); ***G10L 13/00*** (2013.01); ***G10L 13/033*** (2013.01); ***G10L 15/02*** (2013.01); ***G10L 15/063*** (2013.01); ***G10L 15/16*** (2013.01); ***G10L 15/26*** (2013.01); ***G10L 21/003*** (2013.01);

(Continued)

(58) **Field of Classification Search**

CPC G10L 15/02; G10L 15/26; G10L 13/00; G10L 15/063; G10L 2021/0135; G10L 21/007; G10L 21/003; G10L 21/013; G10L 21/01; G10L 15/16; G10L 13/033

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

10,176,819 B2 * 1/2019 Sun G10L 15/187
10,186,251 B1 * 1/2019 Mohammadi G10L 15/26

(Continued)

OTHER PUBLICATIONS

Aryal et al. "Articulatory-based conversion of foreign accents with deep neural networks." Sixteenth Annual Conference of the International Speech Communication Association (Year: 2015).*

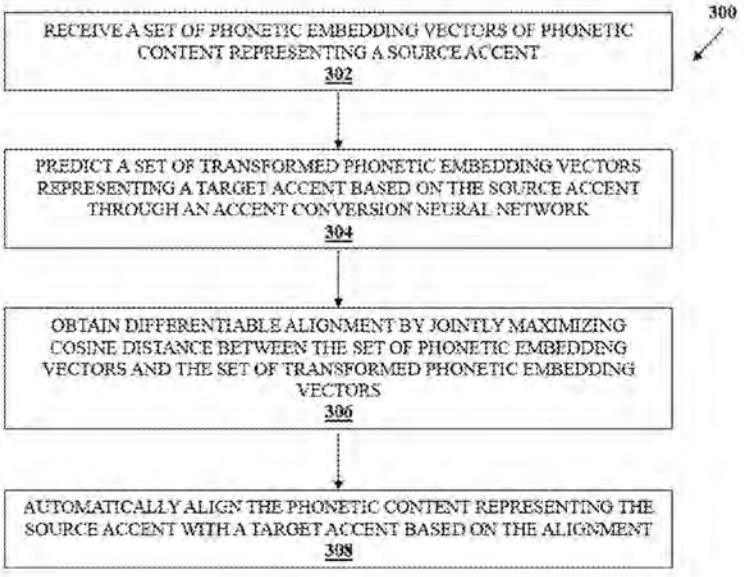
(Continued)

Primary Examiner — Samuel G Neway
(74) *Attorney, Agent, or Firm* — Troutman Pepper Hamilton Sanders LLP

(57) **ABSTRACT**

The disclosed technology relates to methods, accent conversion systems, and non-transitory computer readable media for real-time accent conversion. In some examples, a set of phonetic embedding vectors is obtained for phonetic content representing a source accent and obtained from input audio data. A trained machine learning model is applied to the set of phonetic embedding vectors to generate a set of transformed phonetic embedding vectors corresponding to phonetic characteristics of speech data in a target accent. An alignment is determined by maximizing a cosine distance between the set of phonetic embedding vectors and the set of transformed phonetic embedding vectors. The speech data is then aligned to the phonetic content based on the determined alignment to generate output audio data representing the target accent. The disclosed technology transforms phonetic characteristics of a source accent to match the target accent more closely for efficient and seamless accent conversion in real-time applications.

20 Claims, 3 Drawing Sheets



(51) **Int. Cl.**

G10L 15/06	(2013.01)
G10L 15/16	(2006.01)
G10L 15/26	(2006.01)
G10L 21/003	(2013.01)
G10L 21/01	(2013.01)
G10L 21/013	(2013.01)

(52) **U.S. Cl.**
CPC ***G10L 21/013*** (2013.01); *G10L 21/01* (2013.01); *G10L 2021/0135* (2013.01)

(56) **References Cited**

U.S. PATENT DOCUMENTS

2004/0148161	A1*	7/2004	Das	G10L 21/00 704/E21.001
2021/0193160	A1*	6/2021	Wang	G10L 25/27
2022/0122579	A1*	4/2022	Biadsy	G10L 25/30
2022/0358903	A1*	11/2022	Serebryakov	G10L 13/033
2023/0223006	A1*	7/2023	Fan	H04M 1/72403 704/231
2023/0335107	A1*	10/2023	Zhao	G10L 15/16
2023/0352001	A1*	11/2023	Carmiel	G10L 13/10

OTHER PUBLICATIONS

Liu et al., "End-to-end accent conversion without using native utterances." ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE (Year: 2020).*

* cited by examiner

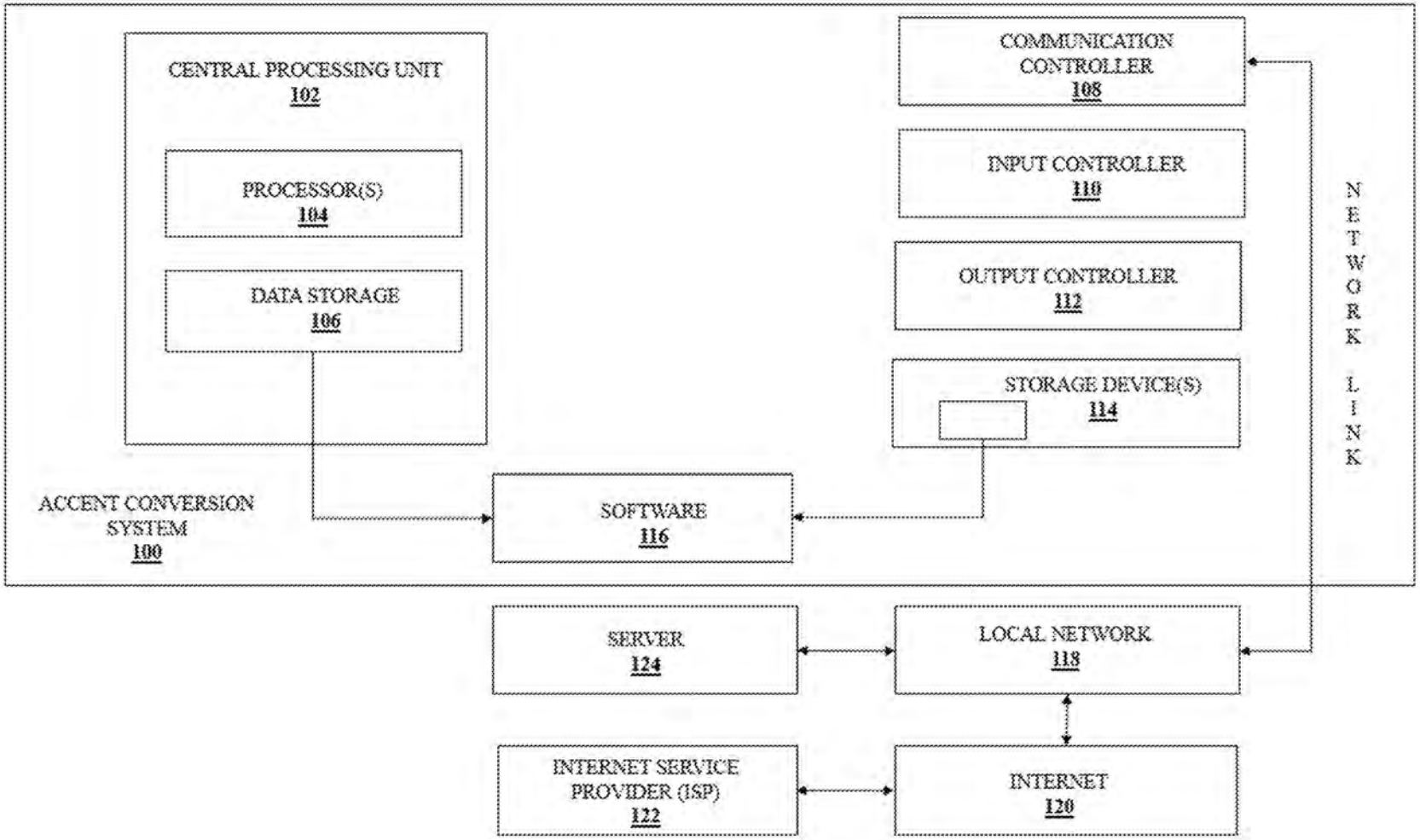

CENTRAL PROCESSING UNIT
102
PROCESSOR(S)
104
DATA STORAGE
106
ACCENT CONVERSION SYSTEM
100
COMMUNICATION CONTROLLER
108
INPUT CONTROLLER
110
OUTPUT CONTROLLER
112
STORAGE DEVICE(S)
114
SOFTWARE
116
NETWORK LINK
SERVER
124
LOCAL NETWORK
118
INTERNET SERVICE PROVIDER (ISP)
122
INTERNET
120

FIG. 1

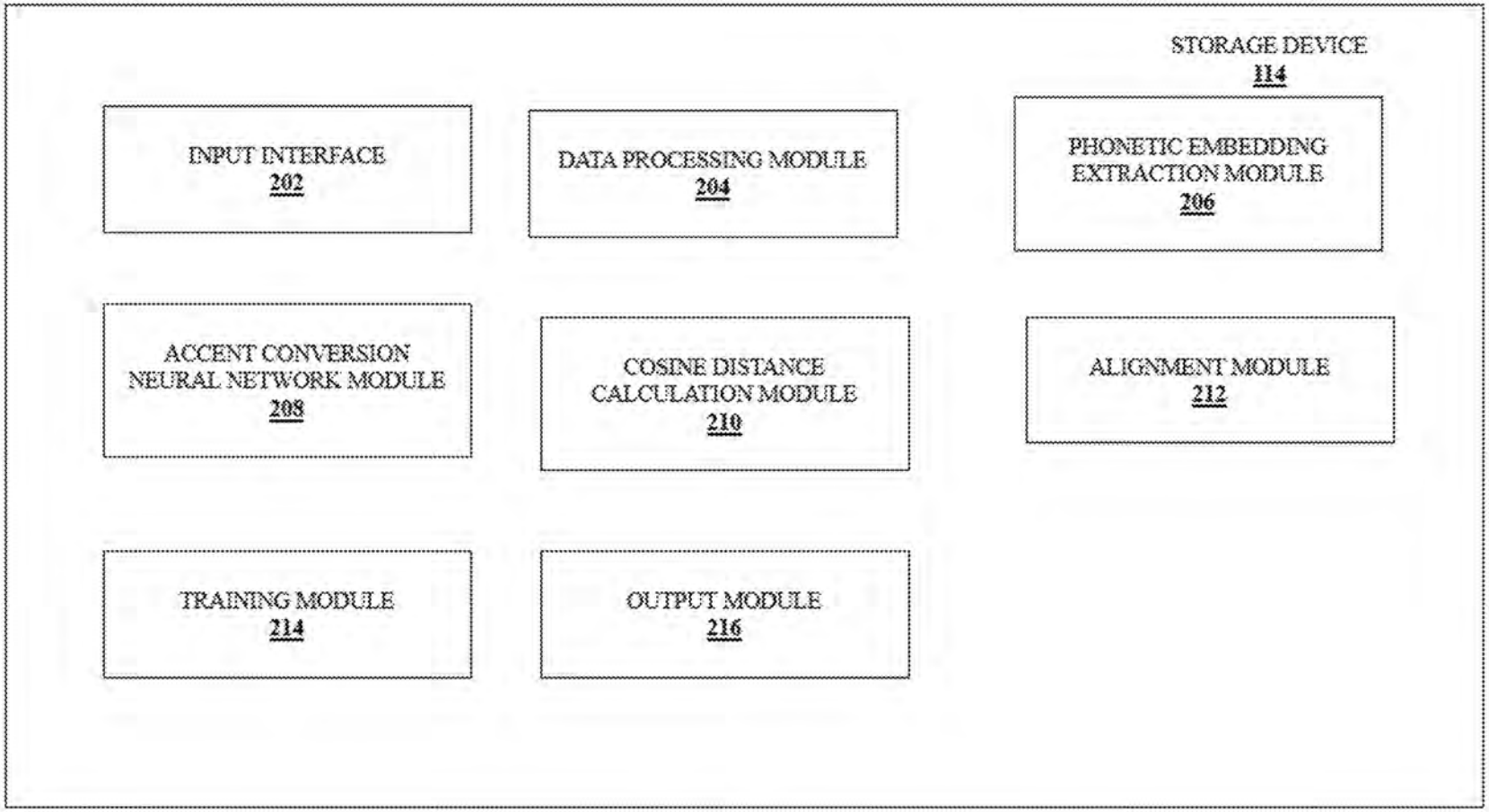

STORAGE DEVICE
114
INPUT INTERFACE
202
DATA PROCESSING MODULE
204
PHONETIC EMBEDDING EXTRACTION MODULE
206
ACCENT CONVERSION NEURAL NETWORK MODULE
208
COSINE DISTANCE CALCULATION MODULE
210
ALIGNMENT MODULE
212
TRAINING MODULE
214
OUTPUT MODULE
216

FIG. 2

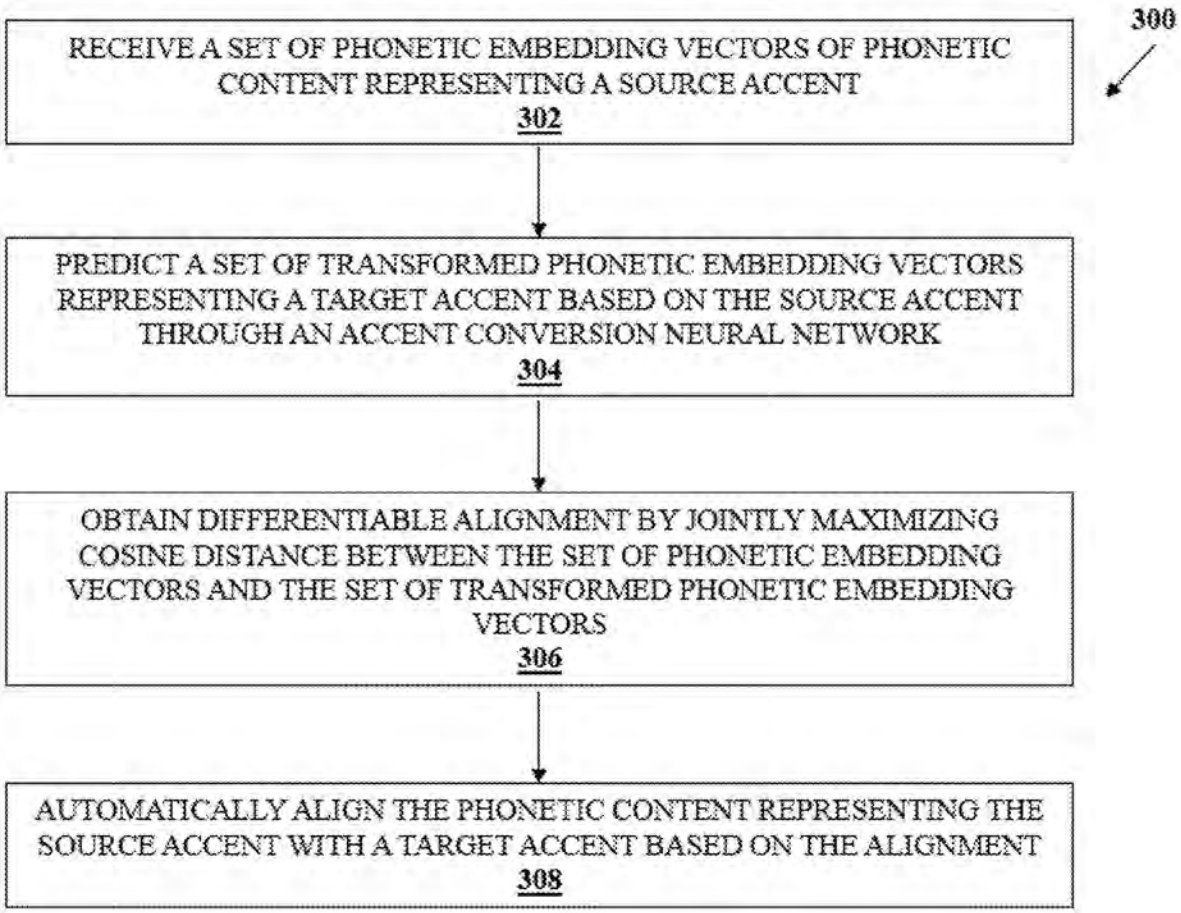

300
RECEIVE A SET OF PHONETIC EMBEDDING VECTORS OF PHONETIC CONTENT REPRESENTING A SOURCE ACCENT
302
PREDICT A SET OF TRANSFORMED PHONETIC EMBEDDING VECTORS REPRESENTING A TARGET ACCENT BASED ON THE SOURCE ACCENT THROUGH AN ACCENT CONVERSION NEURAL NETWORK
304
OBTAIN DIFFERENTIABLE ALIGNMENT BY JOINTLY MAXIMIZING COSINE DISTANCE BETWEEN THE SET OF PHONETIC EMBEDDING VECTORS AND THE SET OF TRANSFORMED PHONETIC EMBEDDING VECTORS
306
AUTOMATICALLY ALIGN THE PHONETIC CONTENT REPRESENTING THE SOURCE ACCENT WITH A TARGET ACCENT BASED ON THE ALIGNMENT
308

FIG. 3

SYSTEM AND METHOD FOR AUTOMATIC ALIGNMENT OF PHONETIC CONTENT FOR REAL-TIME ACCENT CONVERSION

This application claims priority to U.S. Provisional Patent Application Ser. No. 63/510,487, filed Jun. 27, 2023, which is hereby incorporated herein by reference in its entirety.

FIELD

This technology generally relates to audio analysis and, more particularly, to methods and systems for automatic alignment of phonetic content for real-time accent conversion.

BACKGROUND

Real-time accent conversion relates to the process of transforming speech from one accent to another accent in real-time. For instance, a speaker with an Indian accent could have their speech automatically converted into an American accent while they are speaking. This transformation process involves aligning phonetically dissimilar audio of two accents, which can be challenging due to the unique pronunciation styles of each speaker and associated accent.

One approach to aligning two audio sequences uses a dynamic time warping (DTW) algorithm. DTW finds optimal temporal alignment of two sequences by stretching or compressing them in time. However, DTW has limitations, such as being non-differentiable and not providing gradient information. As a result, training an accent conversion model of an accent conversion system using DTW requires two separate steps. The first step involves using DTW to align the audio of the two accents and the second step involves training the accent conversion model using the aligned data. This approach can limit the overall performance of the accent conversion system since the accent conversion model can only learn from the aligned data and not from the original audio.

Non-differentiability also is a significant issue that makes it difficult to train an accent conversion model effectively using DTW, thereby limiting its performance in real-world scenarios. Specifically, the non-differentiability of DTW makes it challenging to optimize current accent conversion systems using gradient-based methods, which are widely used in deep learning models. This limitation can lead to inaccuracies and errors in the accent conversion process and resulting poor-quality audio signals.

Non-monotonicity and instability are other significant issues that lead to alignment errors and negatively impact the accuracy of current accent conversion systems. Non-monotonicity refers to the fact that some alignment algorithms, including DTW, do not always guarantee that the alignment will be strictly increasing in time. This may lead to alignment errors and result in inaccurate accent conversions. Instability refers to the fact that the alignment algorithm may produce different results when the input signals are slightly perturbed, leading to inconsistencies in the accent conversion process.

Other deficiencies of existing accent conversion methods is that they do not handle complex accents that deviate significantly from the data used to train the accent conversion model. In such cases, current accent conversion systems may produce inaccurate or inconsistent results. Additionally, existing accent conversion methods are not able to capture the nuances and variations of different accents accurately, which may affect the naturalness and intelligibility of the converted speech. Furthermore, existing accent conversion methods require a significant amount of training data, which may be a challenge to collect and annotate, limiting the scalability of current systems and making it challenging for current systems to adapt to new accents or languages.

These and other limitations make it challenging to develop and deploy effective real-time accent conversion models and systems to accurately convert accented speech in different audio signals. Accordingly, current accent conversion systems have limited performance, accuracy, and effectiveness for real-time accent conversion.

BRIEF DESCRIPTION OF THE DRAWINGS

The disclosed technology is illustrated by way of example and not limitation in the accompanying figures, in which like references indicate similar elements:

FIG. **1** is a block diagram of an exemplary network environment that includes an accent conversion system;

FIG. **2** is a block diagram of an exemplary storage device of the accent conversion system of FIG. **1**; and

FIG. **3** is a flowchart of an exemplary method for automatic alignment of phonetic content for real-time accent conversion.

DETAILED DESCRIPTION

Examples described below may be used to provide a method, a device (e.g., non-transitory computer readable medium), an apparatus, and/or a system for automatic alignment of phonetic content for real-time accent conversion. Although the technology has been described with reference to specific examples, various modifications may be made to these examples without departing from the broader spirit and scope of the various embodiments of the technology described and illustrated by way of the examples herein.

With this technology, a set of phonetic embedding vectors that represent a source accent is received and a set of transformed phonetic embedding vectors that represent a target accent is predicted using a machine learning model (e.g., an accent conversion neural network). The disclosed technology achieves alignment by maximizing the cosine distance between the two sets of phonetic embedding vectors. Based on this alignment, the phonetic content of the source accent is automatically aligned with the target accent.

Accordingly, this technology enables an accent conversion neural network, for example, to accurately transform the phonetic characteristics of the source accent to closely match the target accent, allowing for efficient and seamless accent conversion in real-time applications. The technology enables efficient and real-time accent conversion, thereby facilitating the processing of speech data in various practical applications such as voice assistants, language learning tools, and speech recognition systems.

In some examples, the disclosed technology may include several components such as audio input, pre-trained phonetic embedding vectors, a neural network model, a gradient-based optimization algorithm, loss functions, training data, and/or a user interface, each of which is described and illustrated in detail below. The technology may include obtaining audio recordings of two different accents and using pre-trained phonetic embedding vectors to quantify the similarity between those accents. An accent conversion neural network machine learning model may then be employed to predict the set of phonetic embedding vectors representing the target accent.

To achieve alignment, a gradient-based optimization algorithm may be utilized to automatically derive an alignment between the two sets of phonetic embedding vectors. The alignment may ensure stability and monotonicity by incorporating various loss functions such as Loss1, Loss2, and Loss3. The gradient-based optimization algorithm may efficiently calculate the alignment by taking advantage of the mathematical properties of the phonetic embedding vectors, which may have a unit norm. The similarity between any two phonetic embedding vectors may be expressed using the cosine distance, which may range between −1 and 1. The user interface may provide real-time feedback on the converted speech, enabling users to adjust settings for optimal performance. Thus, the disclosed technology overcomes limitations in existing solutions and provides gradients for more efficient and effective training of the accent conversion system **100**.

Referring now to FIG. **1**, a block diagram of an exemplary network environment that includes an accent conversion system **100** is illustrated. The accent conversion system **100** in this example is configured for automatic alignment of phonetic content for real-time accent conversion and includes processor(s) **104**, which are designed to process instructions (e.g., computer readable instructions (i.e., code)) stored on the storage device(s) **114** (e.g., a non-transitory computer readable medium) of the accent conversion system **100**. By processing the stored instructions, the processor(s) **104** may perform one or more of the steps and/or functions disclosed herein, such as with reference to FIG. **3** for example.

The storage device(s) **114** may be optical storage device(s), magnetic storage device(s), solid-state storage device(s) (e.g., solid-state disks (SSDs)) or non-transitory storage device(s), another type of memory, and/or a combination thereof, for example, although other types of storage device(s) can also be used. The storage device(s) **114** may contain software **116**, which is a set of instructions (i.e., program code). Alternatively, instructions may be stored in one or more remote storage devices, for example storage devices (e.g., hosted by a server **124**) accessed over a local network **118** or the Internet **120** via an Internet Service Provider (ISP) **122**.

The accent conversion system **100** also includes an operating system and microinstruction code in some examples, one or both of which can be hosted by the storage device(s) **114**. The various processes and functions described herein may either be part of the microinstruction code and/or program code (or a combination thereof), which is executed via the operating system. The accent conversion system **100** also may have data storage **106**, which along with the processor(s) **104** form a central processing unit (CPU) **102**, an input controller **110**, an output controller **112**, and/or a communication controller **108**. A bus (not shown) may operatively couple components of the accent conversion system **100**, including processor(s) **104**, data storage **106**, storage device(s) **114**, input controller **110**, output controller **112**, and/or any other devices (e.g., a network controller or a sound controller).

Output controller **112** may be operatively coupled (e.g., via a wired or wireless connection) to a display device (e.g., a monitor, television, mobile device screen, touch-display, etc.) in such a fashion that output controller **112** can transform the display on the display device (e.g., in response to the execution of module(s)). Input controller **110** may be operatively coupled (e.g., via a wired or wireless connection) to an input device (e.g., mouse, keyboard, touchpad scroll-ball, touch-display, etc.) in such a fashion that input can be received from a user of the accent conversion system **100**.

The communication controller **108** is coupled to a bus (not shown) in some examples and provides a two-way coupling through a network link to the Internet **120** that is connected to a local network **118** and operated by an ISP **122**, which provides data communication services to the Internet. The network link typically provides data communication through one or more networks to other data devices. For example, a network link may provide a connection through local network **118** to a host computer and/or to data equipment operated by the ISP **122**. A server **124** may transmit requested code for an application through the Internet **120**, ISP **122**, local network **118** and/or communication controller **108**.

The accent conversion system **100** is illustrated in FIG. **1** with all components as separate devices for ease of identification only. One or more of the components of the accent conversion system **100** in other examples may be separate devices (e.g., a personal computer connected by wires to a monitor and mouse), may be integrated in a single device (e.g., a mobile device with a touch-display, such as a smartphone or a tablet), or any combination of devices (e.g., a computing device operatively coupled to a touch-screen display device, a plurality of computing devices attached to a single display device and input device, etc.). The accent conversion system **100** also may be one or more servers, for example a farm of networked or distributed servers, a clustered server environment, or a cloud network of computing devices. Other network topologies can also be used in other examples.

Referring now to FIG. **2**, a block diagram of an exemplary one of the storage device(s) **114** of the accent conversion system **100** is illustrated. The storage device(s) **114** may include an input interface **202**, a data processing module **204**, a phonetic embedding extraction module **206**, an accent conversion neural network module **208**, a cosine distance calculation module **210**, an alignment module **212**, a training module **214**, and/or an output module **216**, although other types and/or number of modules can also be used in other examples.

The input interface **202** may serve as an interface through which the accent conversion system **100** receives input data and may allow for the input of the phonetic content representing a source accent, which may be necessary for the accent alignment and conversion process. The phonetic content may be in the form of speech and/or audio data or any other representation that captures the phonetic characteristics of the source accent.

The input interface **202** may include various components or functionalities to facilitate the input process and may include hardware components like microphones or audio interfaces for capturing real-time speech data. Alternatively, input interface **202** may include a software interface that allows for the input of pre-recorded speech data or textual representations of the phonetic content, and other types of input interfaces can also be used in other examples.

Accordingly, the input interface **202** may facilitate the receipt by the accent conversion system **100** of the necessary data to initiate the accent alignment and conversion process described and illustrated herein. The input interface **202** may be the initial point of interaction between a user (e.g., a user computing device) or external systems and the accent conversion system **100**. The input data provided through the input interface **202** may serve as the foundation for subse-

quent processing and analysis within the accent conversion system **100**, as described and illustrated in detail below.

The data processing module **204** may handle the input data received from the input interface **202**. The data processing module **204** may employ techniques such as signal processing, statistical analysis, or machine learning algorithms to extract meaningful information from the input data. This information may include phonetic features, linguistic characteristics, and/or other relevant parameters that contribute to the alignment and conversion process. The data processing module **204** may involve data integration from multiple sources or data fusion techniques to combine different types of input data, enabling a more comprehensive analysis and alignment. The processed data from the data processing module **204** may be then passed on to subsequent modules within the accent conversion system **100**, such as the phonetic embedding extraction module **206** or the accent conversion neural network module **208**, for example, for further analysis and transformation.

The phonetic embedding extraction module **206** may extract phonetic embedding vectors from the phonetic content representing the source accent. The phonetic embedding extraction module **206** may capture and represent the phonetic characteristics of the input speech or audio data in a numerical format. The phonetic embedding vectors may encode essential information about phonemes, speech sounds, or other relevant phonetic units present in the input speech or audio data representing the source accent.

The phonetic embedding extraction module **206** may utilize various techniques, such as deep learning models, feature extraction algorithms, or linguistic analysis methods, to convert the acoustic or linguistic properties of the input speech or audio data into meaningful phonetic embedding vectors. These phonetic embedding vectors are typically high-dimensional numerical representations that capture the distinguishing phonetic features and patterns in the input speech or audio data representing the source accent.

To extract the phonetic embedding vectors, the phonetic embedding extraction module **206** may analyze different aspects of the input speech data, such as spectral features, pitch, formant frequencies, or other acoustic properties. The phonetic embedding extraction module **206** may also consider linguistic information, such as phoneme sequences or linguistic features derived from the input speech data.

The extraction process performed by the phonetic embedding extraction module **206** may involve mapping the input speech data into a latent space where phonetic similarities and differences are captured. This latent space representation may enable the subsequent alignment and conversion step(s) to compare and manipulate the phonetic content effectively, as explained in more detail below.

The extracted phonetic embedding vectors from the phonetic embedding extraction module **206** may serve as a compact and informative representation of the phonetic content in the input speech data associated with the source accent. These vectors may then be utilized by subsequent modules, such as the accent conversion neural network module **208** or the cosine distance calculation module **210**, to perform alignment, conversion, and/or distance computations.

In particular, the accent conversion neural network module **208**, may predict the transformed phonetic embedding vectors that represent the target accent based on the source accent. The accent conversion neural network module **208** may utilize machine learning models including deep learning techniques, specifically neural networks, to learn the mapping between the phonetic embedding vectors of the input speech data in the source accent and the corresponding transformed phonetic embedding vectors that embody the phonetic characteristics of speech data in the target accent. The accent conversion neural network module **208** may leverage the power of neural networks to capture complex patterns and relationships within the phonetic data.

The accent conversion neural network module **208** can include multiple layers, including an encoder layer and a decoder layer. The encoder layer may take the phonetic embedding vectors associated with the source accent as input and encode them into a latent representation, effectively capturing the unique phonetic features of the input speech data representing the source accent. The decoder layer may then decode this latent representation to generate the transformed phonetic embedding vectors that represent the target accent.

During the training phase, the accent conversion neural network module **208** may learn to predict the transformed phonetic vectors by adjusting the internal parameters based on a labeled dataset. This dataset may include paired samples of source accent phonetic embedding vectors and corresponding target accent phonetic embedding vectors. By iteratively adjusting the network parameters, the accent conversion neural network module **208** may optimize its predictions to minimize the difference between the predicted transformed vectors and the target accent vectors.

In some examples of the real-time accent conversion process described and illustrated by way of the examples herein, the accent conversion neural network module **208** may take the phonetic embedding vectors of the source accent as input and pass them through the accent conversion neural network. The internal computations of the neural network and learned transformations may enable the accent conversion neural network module **208** to generate the transformed phonetic embedding vectors that represent the target accent. The transformed phonetic embedding vectors may capture the phonetic characteristics and nuances of the target accent, allowing for a seamless conversion from the source accent to the target accent. Deep learning capabilities of the accent conversion neural network module **208** may make it capable of capturing subtle accent-specific details, resulting in accurate and effective accent conversion.

The cosine distance calculation module **210** in some examples is configured to calculate the cosine distance between the set of phonetic embedding vectors and the set of transformed phonetic embedding vectors. The cosine distance may measure similarity between two vectors that considers both their direction and magnitude. By jointly maximizing the cosine distance between the phonetic embedding vectors and the transformed phonetic embedding vectors, the cosine distance calculation module **210** may facilitate the alignment process described herein.

The cosine distance calculation module **210** is configured to normalize both sets of phonetic embedding vectors to have a unit norm. Normalization may involve scaling the phonetic embedding vectors to have a magnitude or length of one, while preserving their relative directions. This normalization may ensure that the phonetic embedding vectors are on a consistent scale and eliminate the influence of their magnitudes in the cosine distance calculation.

Once the phonetic embedding vectors are normalized, the cosine distance calculation module **210** may compute the dot product of the normalized phonetic embedding vectors. The dot product may measure the similarity of the phonetic embedding vectors based on their directions. By taking the dot product of the normalized phonetic embedding vectors, the cosine distance calculation module **210** may calculate

the cosine distance between them, which provides a measure of alignment between the phonetic embedding vectors and the transformed phonetic embedding vectors. Maximizing the cosine distance may jointly align the phonetic embedding vectors in a way that minimizes their dissimilarity and maximizes their similarity.

The computed cosine distance may also serve as a feedback signal for the alignment module **212**, enabling it to optimize and refine the alignment process. By maximizing the cosine distance, the alignment module **212** may achieve an improved alignment, enabling accurate and effective accent conversion. Accordingly, the alignment module **212** in some examples aligns the phonetic content of the input speech data associated with the source accent with the target accent based on the alignment obtained through the cosine distance maximization.

Once the cosine distance between the phonetic embedding vectors and the transformed phonetic embedding vectors is calculated, the alignment module **212** may utilize this information to perform an alignment process. The alignment module **212** may automatically align the phonetic content of the input speech data representing the source accent with the target accent to closely match each other. The alignment module **212** may operate at a frame-level granularity in some examples, aligning individual frames of the input speech data associated with the source accent with corresponding frames of the target accent. The fine-grained alignment may allow for relatively precise matching of the phonetic content between accents, capturing temporal characteristics of speech.

To achieve alignment, the alignment module **212** may employ various techniques such as time-warping functions. These functions may enable the temporal alignment of the phonetic content by stretching or compressing the frames of the speech data representing the source accent to match the corresponding frames of output speech data representing the target accent. The temporal alignment may ensure that the phonetic content is properly synchronized between the accents.

By performing automatic alignment, the alignment module **212** may facilitate transformation of the phonetic characteristics of the input speech data in the source accent to closely match those of the target accent. This alignment process may ensure that important phonetic features are preserved while adapting the phonetic content to the desired target accent. Thus, the alignment module **212** allows for seamless and efficient conversion of accents during speech processing and may ensure that the converted speech maintains the natural flow and rhythm while accurately reflecting the desired target accent.

The training module **214** is configured to train the accent conversion neural network by iteratively adjusting one or more phonetic parameters based on the alignment achieved through maximizing the cosine distance. During the training process, the training module **214** may use a dataset comprising paired samples of source accent phonetic embedding vectors and target accent phonetic embedding vectors. These paired samples may serve as the training data for the accent conversion neural network.

The training module **214** may employ a gradient-based optimization algorithm to optimize the joint maximization of the cosine distance. The gradient-based optimization algorithm may iteratively update the phonetic parameters of the accent conversion neural network based on the calculated gradients of a loss function, aiming to minimize the discrepancy between the predicted transformed phonetic embedding vectors and the target accent vectors.

By adjusting the phonetic parameters, the training module **214** may ensure that the accent conversion neural network learns to generate accurate and meaningful transformations of phonetic embedding vectors from the source accent to the target accent. The training process may allow the accent conversion neural network to capture the underlying patterns and relationships between the accents, enabling it to perform accurate accent conversion.

The training module **214** may fine-tune the ability of the accent conversion neural network to align and convert accents effectively. By continually updating the network parameters, the training module **214** may improve the accent conversion neural network performance and enhance its capability to produce high-quality transformed phonetic embedding vectors that closely match the target accent.

The training phase may be performed before deploying the accent conversion system **100** for real-time accent conversion and may involve multiple iterations and the adjustment of various phonetic parameters to achieve optimal performance. The training module **214** may enable the accent conversion neural network to learn and improve its accent conversion capabilities, leading to more accurate and reliable results in real-time accent conversion scenarios.

Once the alignment module **212** aligns the phonetic content of the source accent with the target accent, the aligned phonetic content may be passed to the output module **216** for further processing. The output module **216** in some examples is configured to generate speech output data that closely resembles the target accent while preserving the original linguistic content. The output module **216** may incorporate techniques such as prosody modeling, intonation adjustment, and/or accent-specific acoustic modeling for high quality, natural sounding, accurate and fluent speech production in the target accent.

The output module **216** may offer options for adjusting the speech characteristics, such as speech rate, pitch, or gender, to further customize the converted speech output based on user preferences or application requirements, for example. The output module **216** may deliver a seamless and intelligible speech output to reflect the desired target accent. By leveraging advanced speech synthesis techniques and models, the output module **216** may provide an accurate representation of the converted accent, allowing users to hear the converted speech output with the intended target accent in real-time or on-demand.

Referring now to FIG. **3**, a flowchart of an exemplary method **300** for automatic alignment of phonetic content for real-time accent conversion is illustrated. In some examples, the method **300** may be implemented as a software application (e.g., software **116** executed by the central processing unit **102**) or a module within a larger speech processing system. The software application or module may receive input audio data, perform automatic alignment, accent conversion, and provide the converted speech output in real-time, as explained in detail below

In step **302** in this example, the accent conversion system **100** receives a set of phonetic embedding vectors of phonetic content representing a source accent. The phonetic content is associated with speech in the source accent as represented within audio data from which the phonetic embedding vectors are generated. The phonetic content representing the source accent can be associated with audio data captured (e.g., via a microphone) or obtained by the accent conversion system **100**. Audio embeddings capture audio data, including speech, as numerical vectors, incorporating acoustic features and temporal patterns in the audio, for example. Thus, the phonetic embedding vectors in

some examples capture important features related to pronunciation, intonation, and other phonetic aspects of the speech in the source accent.

In some examples, additional phonetic embedding vectors can be used that represent emotions or styles, for example. In this examples, the method **300** may align and convert not only accents but also emotional or stylistic aspects of the source speech, enabling more versatile and expressive accent conversion applications. The phonetic embedding vectors can be generated by a machine learning model (also referred to as an embedding model) trained to generate the phonetic embedding vectors from input audio data (e.g., audio data encapsulating the phonetic content representing the source accent).

In step **304**, the accent conversion system **100** predicts a set of transformed phonetic embedding vectors representing a target accent based on the source accent through a trained accent conversion neural network. The target accent can be selected by a user and/or a stored default accent in some examples. Accordingly, the accent conversion neural network may be trained to predict a set of transformed phonetic embedding vectors that represent the target accent, based on input from the source accent. The accent conversion neural network may be trained by iteratively adjusting one or more parameters, utilizing the alignment achieved through maximizing the cosine distance between the source and target phonetic embedding vectors, which is described and illustrated in detail herein.

In some examples, the accent conversion system **100** trains the accent conversion neural network using a large dataset of aligned phonetic content pairs from multiple source and target accents, which allows the accent conversion neural network to learn a more generalized mapping between different accents, enhancing its accent conversion capabilities. Optionally, the accent conversion system **100** can subsequently preprocess the source and target phonetic embedding vectors (also referred to herein as the set of phonetic embedding vectors and the set of transformed phonetic embedding vectors, respectively) by applying dimensionality reduction techniques, such as Principal Component Analysis (PCA) or t-SNE (t-Distributed Stochastic Neighbor Embedding) to reduce the computational complexity and enhance the alignment accuracy.

In step **306**, the accent conversion system **100** obtains differentiable alignment by jointly maximizing cosine distance between the set of phonetic embedding vectors and the set of transformed phonetic embedding vectors. To ensure accurate alignment, the accent conversion system **100** employs a joint maximization of the cosine distance between the set of phonetic embedding vectors and the set of transformed phonetic embedding vectors, which allows for alignment and enables a smooth and seamless transition between source and target accents. In one example, the joint maximization of the cosine distance between the set of phonetic embedding vectors and the set of transformed phonetic embedding vectors may be performed using a gradient-based optimization algorithm.

Optionally, the phonetic embedding vectors and the predicted set of transformed phonetic embedding vectors can be normalized to have a unit norm, which ensures that the phonetic embedding vectors are scaled to a standardized length, specifically a magnitude of one, using mathematical calculations, such as dividing each component of each of the phonetic embedding vectors by its Euclidean norm or another appropriate norm. Normalization may involve scaling the phonetic embedding vectors to have a unit norm, followed by computing a dot product of the normalized phonetic embedding vectors, for example.

The normalization may create a consistent scale for the phonetic embedding vectors, allowing for effective comparisons and calculations based on their direction or relative positions rather than their magnitudes. In other words, by normalizing the phonetic embedding vectors to a unit norm, their magnitudes are equalized, and the focus may be shifted towards their orientations or relationships.

Thus, a unit norm for the phonetic embedding vectors and the predicted set of transformed phonetic embedding vectors may be used for calculating the cosine distance. The cosine distance is a measure of the angle between two vectors and may be used to quantify their similarity or dissimilarity. The cosine distance calculation may be more accurate and reliable by normalizing the phonetic embedding vectors to have a unit norm, leading to relatively precise alignment. According, in some examples, the accent conversion system **100** calculates the cosine distance between the normalized phonetic embedding vectors and the predicted set of transformed phonetic embedding vectors to facilitate the alignment process and allow for efficient comparison between the source and target accents.

In step **308**, the accent conversion system **100** automatically aligns the phonetic content of the source accent with the target accent based on the differentiable alignment obtained in step **306** to generate output audio data with phonetic content representing the target accent. The alignment of step **308** advantageously guarantees a relatively precise matching of the phonetic characteristics of the speech in the source and target accents, resulting in a highly accurate and natural-sounding accent conversion.

In some examples, the accent conversion system **100** incorporates a language model or a phonetic dictionary to improve the alignment accuracy. The language model or dictionary may provide additional context and phonetic information, enabling better alignment of the phonetic content between the source and target accents. In yet other examples, the accent conversion system **100** may incorporate a feedback loop mechanism that enables iterative refinement of the alignment and accent conversion by continuously comparing the converted phonetic content with the target accent and adjusting the accent conversion neural network parameters accordingly. Thus, the alignment of step **308** may enable end-to-end training of the accent conversion neural network.

Optionally, the accent conversion system **100** may utilize a speaker adaptation module that adapts the accent conversion method **300** based on the specific characteristics of the speaker's voice (i.e., the speaker associated with the speech content of the input audio data from which the set of phonetic embedding vectors of phonetic content is generated), thereby improving the accuracy and naturalness of the converted speech for individual speakers. In some examples of this technology, the alignment of step **308** is about twenty times faster than alignment achieved using dynamic time warping (DTW).

With this technology, an alignment between a set of phonetic embedding vectors representing a source accent and a set of transformed phonetic embedding vectors representing a target accent is advantageously derived for real-time accent conversion. The technology disclosed herein may be differentiable, may provide gradient, and may allow for more efficient and effective training of a real-time accent conversion system **100**.

This technology has numerous practical applications, such as accent modification in speech synthesis, language

learning tools, and cross-accent speech recognition. Moreover, the real-time capability of this technology ensures efficient and seamless accent conversion during speech processing, which enables users to communicate more effectively across different accents. The disclosed technology can be applied to non-native speakers learning a new accent in some implementations. By aligning the phonetic content of the learner's native accent with the target accent, this technology may facilitate accent acquisition and help learners improve their pronunciation and intonation.

In yet other applications, this technology can be used with voice assistants and virtual agents. By automatically aligning and converting accents in real-time, these voice assistant and virtual agent systems may provide a more personalized and natural user experience, which enables effective communication between the user and the voice assistant or virtual agent, regardless of the user's accent. The voice assistants and virtual agents may adapt to different accents, enhancing their ability to understand and respond to users' queries and requests.

This technology may also be applicable to multilingual communication systems, such as call centers or language translation services. Specifically, the disclosed technology enables seamless accent conversion by aligning and converting the accents of both the caller and the recipient, which facilitates smooth communication and overcomes potential barriers caused by diverse accents and thereby improves the overall quality and efficiency of multilingual interactions.

Moreover, this technology is applicable to the media and entertainment industries as it may be utilized to modify the accents of actors or voice-over artists to match specific roles or characters. By automatically aligning and converting accents, this technology enhances the authenticity and consistency of accents portrayed in movies, television shows, and other forms of media, which may improve the overall quality and realism of the content and enhance the viewer's experience.

Further, the automatic alignment of phonetic content described and illustrated by way of the examples herein may also benefit speech recognition and natural language processing (NLP) systems. By converting diverse accents into a common reference accent, this technology may improve the accuracy and performance of such systems, which may enable better understanding and interpretation of spoken input, enhance speech recognition, transcription, and language understanding capabilities, and be particularly useful in applications such as voice dictation, transcription services, and language understanding platforms.

Having thus described the basic concept of the invention, it will be rather apparent to those skilled in the art that the foregoing detailed disclosure is intended to be presented by way of example only and is not limiting. Various alterations, improvements, and modifications will occur and are intended for those skilled in the art, though not expressly stated herein. These alterations, improvements, and modifications are intended to be suggested hereby, and are within the spirit and scope of the invention. Additionally, the recited order of processing elements or sequences, or the use of numbers, letters, or other designations, therefore, is not intended to limit the claimed processes to any order.

What is claimed is:

1. An accent conversion system, comprising an audio interface, memory having instructions stored thereon, and one or more processors coupled to the memory and configured to execute the instructions to:

obtain input audio data via the audio interface;

generate from the input audio data first phonetic embedding vectors for phonetic content representing a source accent;

apply a trained accent conversion neural network to the first phonetic embedding vectors to generate second phonetic embedding vectors corresponding to first phonetic characteristics of speech data in a target accent;

determine a differentiable alignment by jointly maximizing a cosine distance between the first phonetic embedding vectors and the second phonetic embedding vectors; and

align the speech data to the phonetic content based on the differentiable alignment to generate and provide output audio data corresponding to the aligned speech data and representing the target accent.

2. The accent conversion system of claim **1**, wherein the first phonetic embedding vectors represent second phonetic characteristics of input speech in the input audio data in a numerical format and encode one or more of phonetic features, patterns, phonemes, pronunciation, intonation, speech sounds, or phonetic units present in the input speech.

3. The accent conversion system of claim **1**, wherein the accent conversion neural network:

is trained to learn a mapping between the first phonetic embedding vectors and the second phonetic embedding vectors using a labeled dataset comprising paired samples of course accent phonetic embedding vectors and corresponding target accent phonetic embedding vectors; and

comprises an encoder layer configured to encode the first phonetic embedding vectors into a latent representation and a decoder layer configured to decode the latent representation to generate the second phonetic embedding vectors.

4. The accent conversion system of claim **1**, wherein the one or more processors are further configured to execute the instructions to, in order to determine the cosine distance:

normalize the first and second phonetic embedding vectors by scaling the first and second phonetic embedding vectors to have a magnitude of one and preserving a relative direction of the first and second phonetic embedding vectors; and

generate a dot product of the normalized first and second phonetic embedding vectors.

5. The accent conversion system of claim **1**, wherein the one or more processors are further configured to execute the instructions to, in order to generate the output audio data, one or more of align first frames of the speech data with corresponding second frames of the phonetic content, apply one or more of prosody modeling, intonation adjustment, or accent-specific acoustic modeling techniques, or adjust a speech rate, pitch, or gender, wherein the output audio data preserves linguistic content of the input audio data.

6. The accent conversion system of claim **1**, wherein the one or more processors are further configured to execute the instructions to apply a gradient-based optimization algorithm to optimize the joint maximization of the cosine distance.

7. The accent conversion system of claim **1**, wherein the one or more processors are further configured to execute the instructions to apply one or more dimensionality reduction techniques to preprocess the first and second phonetic embedding vectors.

8. A method for automatic alignment of phonetic content for real-time accent conversion, the method implemented by an accent conversion system and comprising:

obtaining a set of phonetic embedding vectors for phonetic content representing a source accent and obtained from input audio data;

applying a trained machine learning model to the set of phonetic embedding vectors to generate a set of transformed phonetic embedding vectors corresponding to phonetic characteristics of speech data in a target accent;

determining an alignment by maximizing a cosine distance between the set of phonetic embedding vectors and the set of transformed phonetic embedding vectors; and

aligning the speech data to the phonetic content based on the determined alignment to generate output audio data representing the target accent.

9. The method of claim **8**, wherein the first phonetic embedding vectors represent second phonetic characteristics of input speech in the input audio data in a numerical format and encode one or more of phonetic features, patterns, phonemes, pronunciation, intonation, speech sounds, or phonetic units present in the input speech.

10. The method of claim **8**, wherein the machine learning model:

is trained to learn a mapping between the set of phonetic embedding vectors and the set of transformed phonetic embedding vectors using a labeled dataset comprising paired samples of course accent phonetic embedding vectors and corresponding target accent phonetic embedding vectors; and

comprises an encoder layer configured to encode the set of phonetic embedding vectors into a latent representation and a decoder layer configured to decode the latent representation to generate the set of transformed phonetic embedding vectors.

11. The method of claim **8**, further comprising, in order to determine the cosine distance:

normalizing the set of phonetic embedding vectors and the set of transformed phonetic embedding vectors by scaling the set of phonetic embedding vectors and the set of transformed phonetic embedding vectors to have a same magnitude and preserving a relative direction of the set of phonetic embedding vectors and the set of transformed phonetic embedding vectors; and

generate a dot product of the normalized the set of phonetic embedding vectors and the set of transformed phonetic embedding vectors.

12. The method of claim **8**, further comprising, in order to generate the output audio data, one or more of aligning first frames of the speech data with corresponding second frames of the phonetic content, applying one or more of prosody modeling, intonation adjustment, or accent-specific acoustic modeling techniques, or adjusting a speech rate, pitch, or gender, wherein the output audio data preserves linguistic content of the input audio data.

13. The method of claim **8**, further comprising applying a gradient-based optimization algorithm to optimize the joint maximization of the cosine distance.

14. The method of claim **8**, further comprising applying one or more dimensionality reduction techniques to preprocess the set of phonetic embedding vectors and the set of transformed phonetic embedding vectors.

15. A non-transitory computer-readable medium comprising instructions that, when executed by at least one processor, cause the at least one processor to:

obtain first phonetic embedding vectors for phonetic content representing a source accent and determined from obtained input audio data;

apply a trained neural network to the first phonetic embedding vectors to generate second phonetic embedding vectors corresponding to phonetic characteristics of speech data in a target accent;

determine an alignment by maximizing a cosine distance between the first phonetic embedding vectors and the second phonetic embedding vectors; and

align the speech data to the phonetic content based on the determined alignment to generate output audio data representing the target accent.

16. The non-transitory computer-readable medium of claim **15**, wherein the first phonetic embedding vectors represent second phonetic characteristics of input speech in the input audio data in a numerical format and encode one or more of phonetic features, patterns, phonemes, pronunciation, intonation, speech sounds, or phonetic units present in the input speech.

17. The non-transitory computer-readable medium of claim **15**, wherein the instructions, when executed by the at least one processor further causes the at least one processor to, in order to determine the cosine distance:

normalize the first and second phonetic embedding vectors by scaling the first and second phonetic embedding vectors to have a magnitude of one and preserving a relative direction of the first and second phonetic embedding vectors; and

generate a dot product of the normalized first and second phonetic embedding vectors.

18. The non-transitory computer-readable medium of claim **15**, wherein the instructions, when executed by the at least one processor further causes the at least one processor to, in order to generate the output audio data, one or more of align first frames of the speech data with corresponding second frames of the phonetic content, apply one or more of prosody modeling, intonation adjustment, or accent-specific acoustic modeling techniques, or adjust a speech rate, pitch, or gender, wherein the output audio data preserves linguistic content of the input audio data.

19. The non-transitory computer-readable medium of claim **15**, wherein the instructions, when executed by the at least one processor further causes the at least one processor to apply a gradient-based optimization algorithm to optimize the joint maximization of the cosine distance.

20. The non-transitory computer-readable medium of claim **15**, wherein the instructions, when executed by the at least one processor further causes the at least one processor to apply one or more dimensionality reduction techniques to preprocess the first and second phonetic embedding vectors.

* * * * *

EXHIBIT D

US011715457B1

(12) **United States Patent**
Golman et al.

(10) **Patent No.: US 11,715,457 B1**
(45) **Date of Patent: Aug. 1, 2023**

(54) **REAL TIME CORRECTION OF ACCENT IN SPEECH AUDIO SIGNALS**

(71) Applicant: **Intone Inc.**, New York, NY (US)

(72) Inventors: **Andrei Golman**, San Francisco, CA (US); **Dmitrii Sadykov**, Yerevan (AM)

(73) Assignee: **Intone Inc.**, New York, NY (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **18/083,727**

(22) Filed: **Dec. 19, 2022**

Related U.S. Application Data

(60) Provisional application No. 63/297,901, filed on Jan. 10, 2022.

(51) **Int. Cl.**

G10L 15/00	(2013.01)
G10L 15/02	(2006.01)
G10L 13/04	(2013.01)
G10L 25/30	(2013.01)
G10L 15/22	(2006.01)
G10L 15/183	(2013.01)
G10L 13/08	(2013.01)
G10L 15/18	(2013.01)

(52) **U.S. Cl.**
CPC ***G10L 15/02*** (2013.01); ***G10L 13/04*** (2013.01); ***G10L 15/22*** (2013.01); ***G10L 25/30*** (2013.01); *G10L 13/08* (2013.01); *G10L 15/183* (2013.01); *G10L 15/1822* (2013.01); *G10L 2015/025* (2013.01)

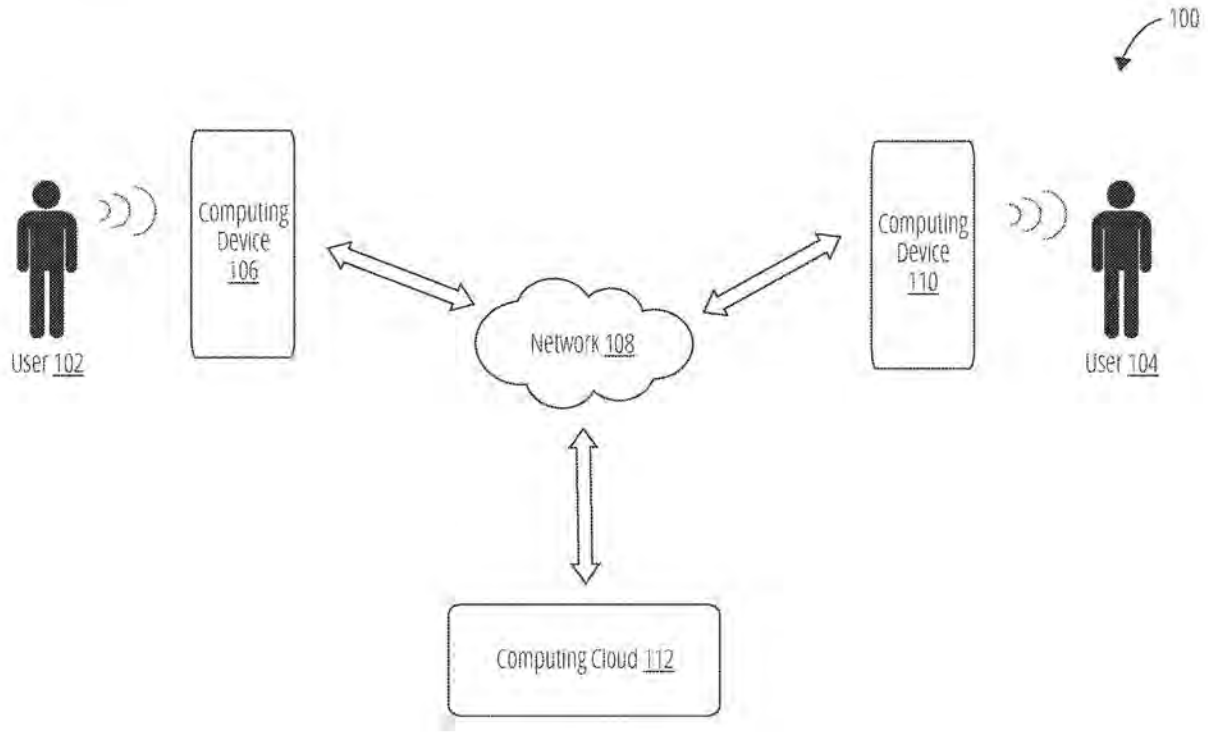
(58) **Field of Classification Search**
CPC combination set(s) only.
See application file for complete search history.

Primary Examiner — Vu B Hang
(74) *Attorney, Agent, or Firm* — Georgiy L. Khayet

(57) **ABSTRACT**

Systems and methods for real-time correction of an accent in a speech audio signal are provided. A method includes dividing the speech audio signal into a stream of input chunks, an input chunk from the stream of input chunks including a pre-defined number of frames of the speech audio signal, extracting, by an acoustic features extraction module from the input chunk and a context associated with the input chunk, acoustic features, the context is a pre-determined number of the frames preceding the input chunk in the stream; extracting, by a linguistic features extraction module from the input chunk and the context, linguistic features, receiving a speaker embedding for a human speaker, providing the speaker embedding, the acoustic features, and the linguistic features to a synthesis module to generate a melspectrogram with a reduced accent, providing the melspectrogram to a vocoder to generate an output chunk of an output audio signal.

20 Claims, 12 Drawing Sheets



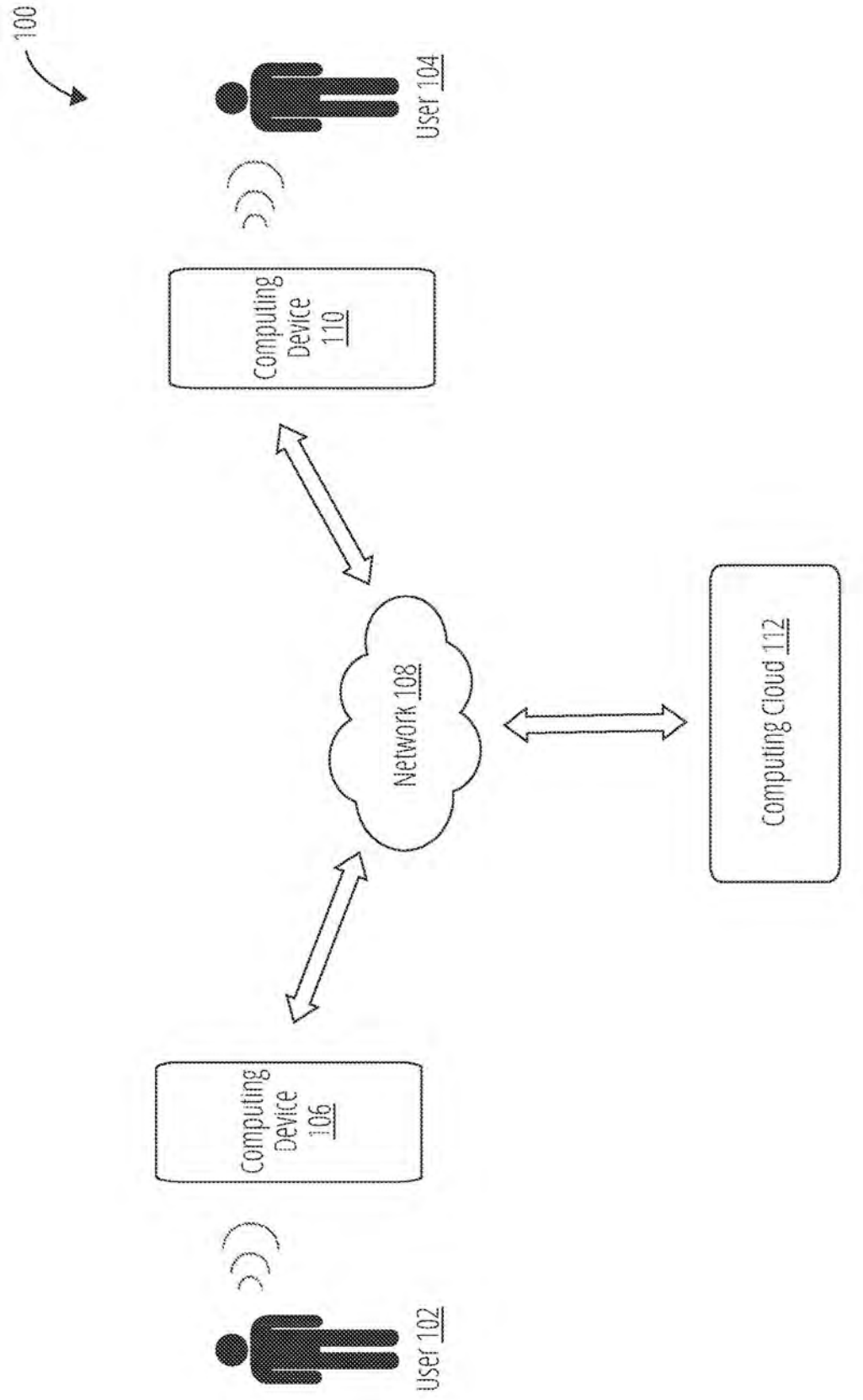

100
User 104
Computing Device 110
Network 108
Computing Cloud 112
Computing Device 106
User 102

FIG. 1

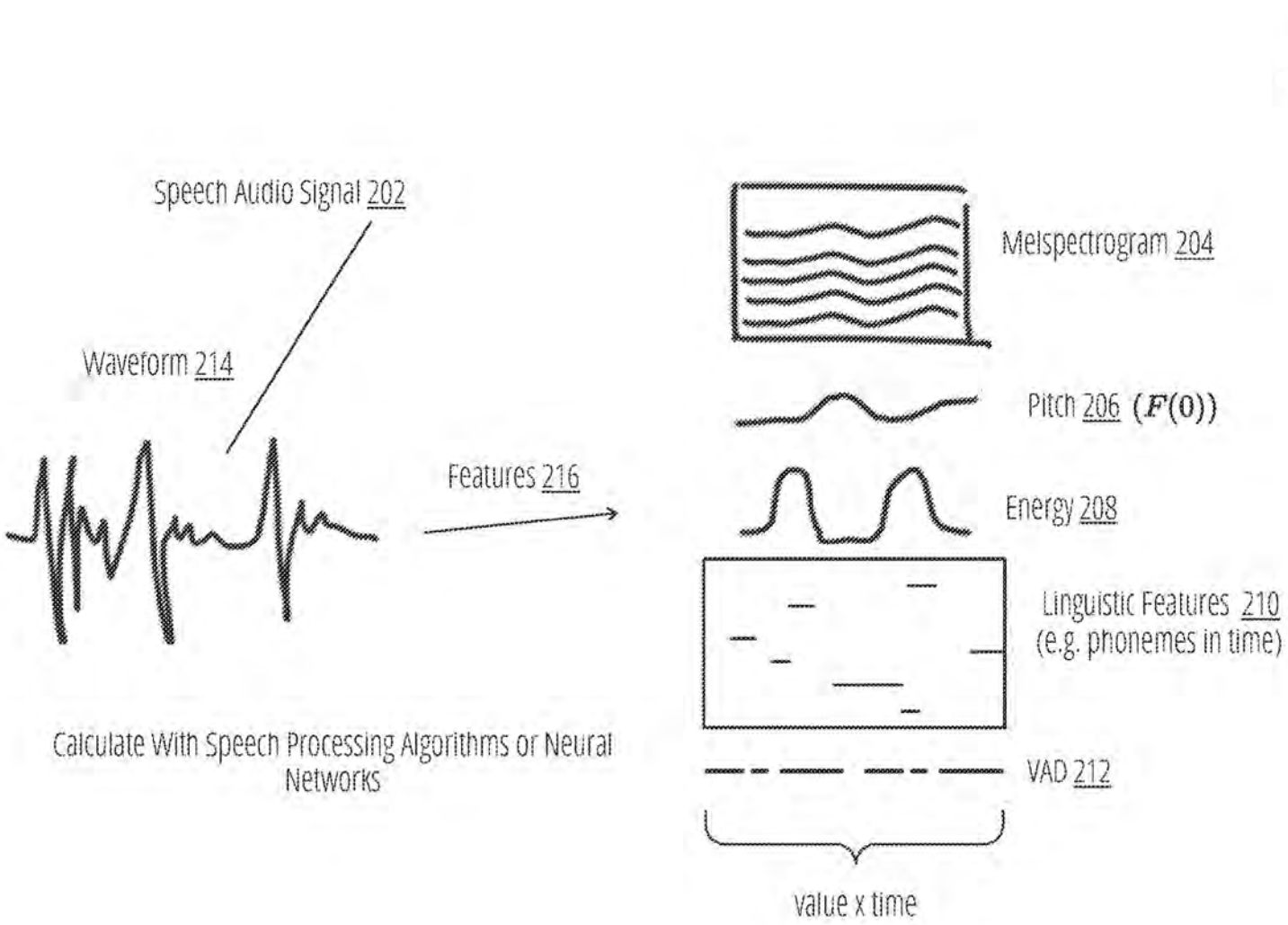

200
Speech Audio Signal 202
Waveform 214
Features 216
Calculate With Speech Processing Algorithms or Neural Networks
Melspectrogram 204
Pitch 206 (F(0))
Energy 208
Linguistic Features 210
(e.g. phonemes in time)
VAD 212
value x time

FIG. 2

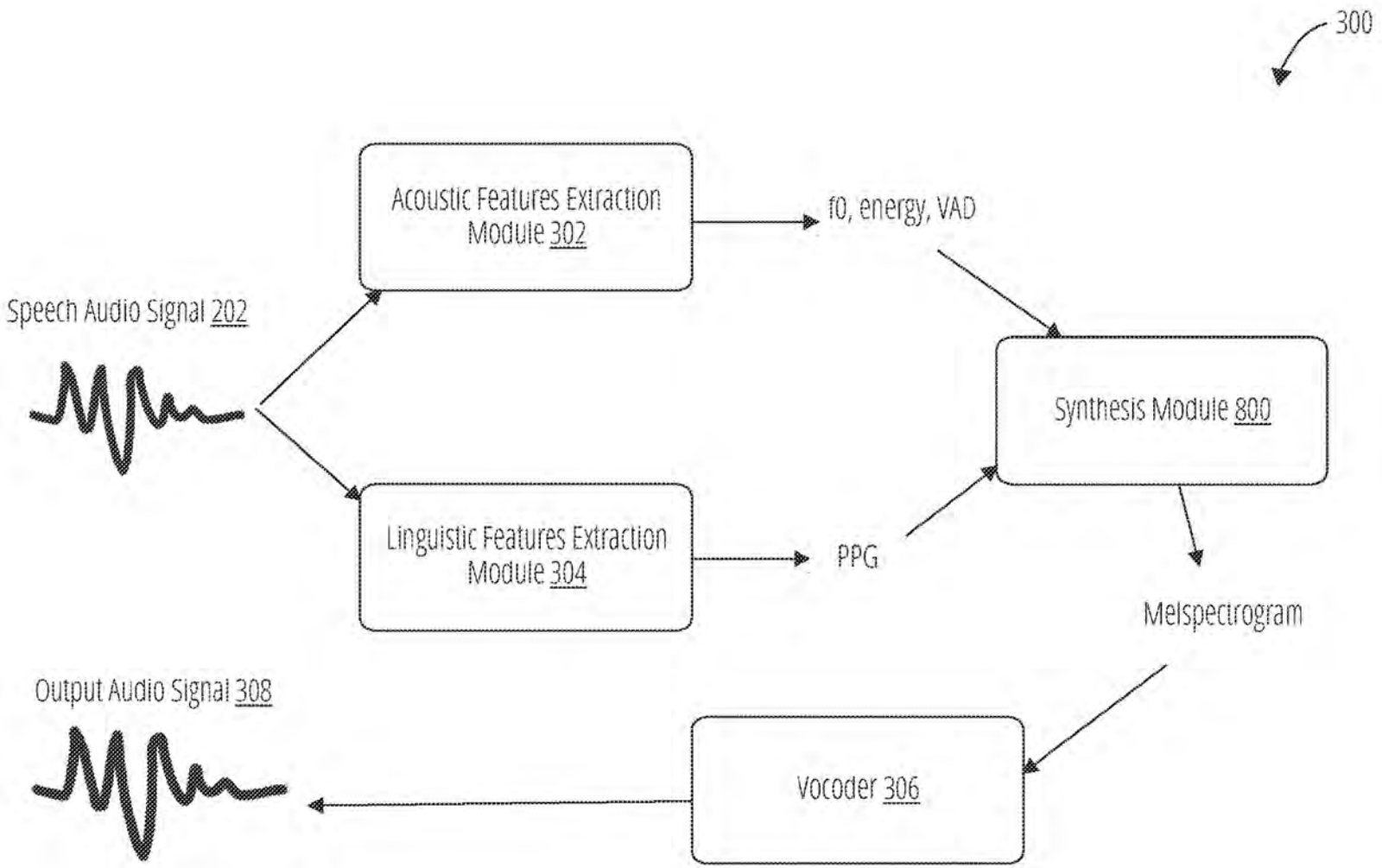

300
Acoustic Features Extraction Module 302
f0, energy, VAD
Speech Audio Signal 202
Synthesis Module 800
Linguistic Features Extraction Module 304
PPG
Melspectrogram
Output Audio Signal 308
Vocoder 306

FIG. 3

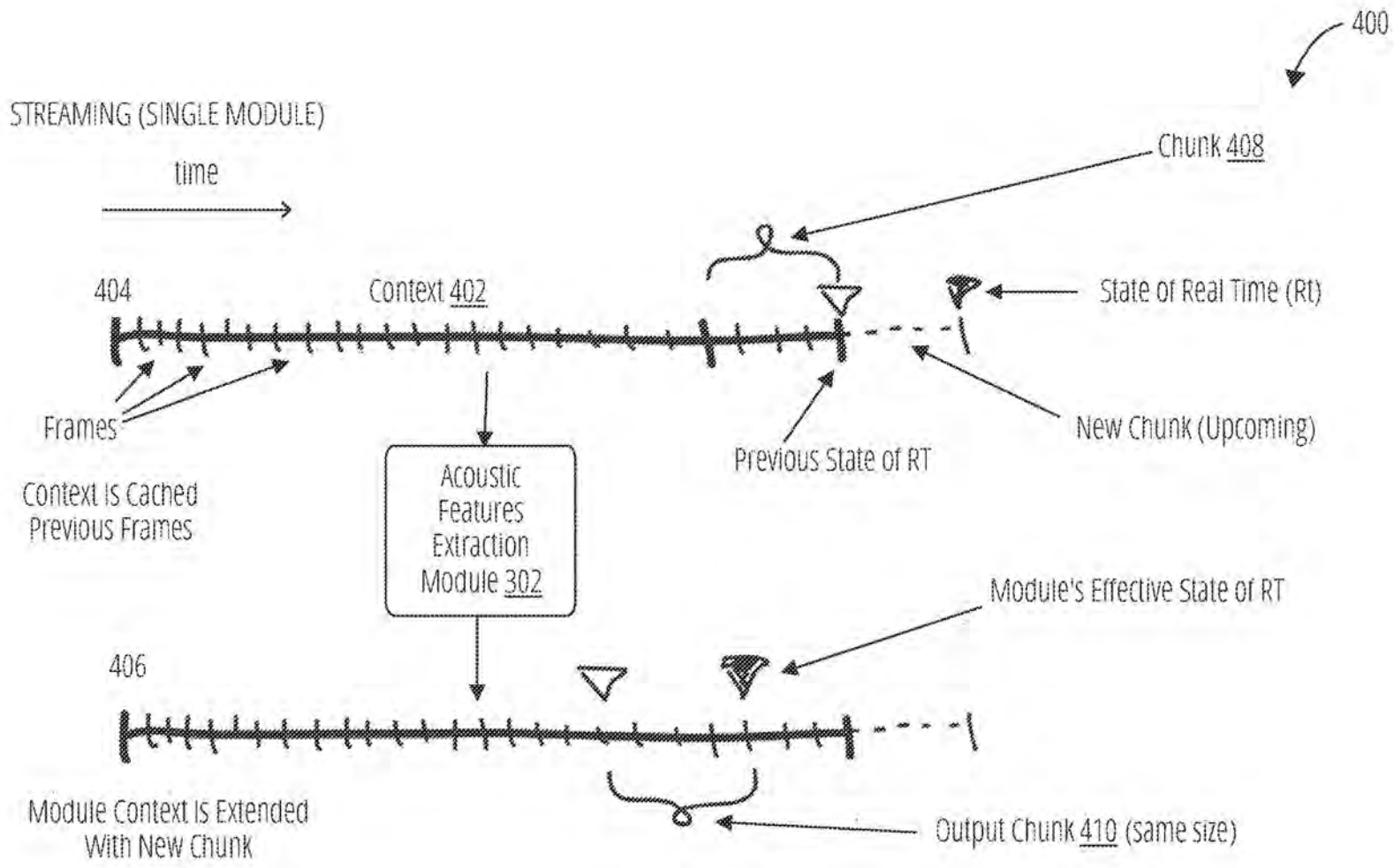

400
STREAMING (SINGLE MODULE)
time
Chunk 408
404
Context 402
State of Real Time (RT)
Frames
Context Is Cached Previous Frames
Acoustic Features Extraction Module 302
Previous State of RT
New Chunk (Upcoming)
Module's Effective State of RT
406
Module Context Is Extended With New Chunk
Output Chunk 410 (same size)

FIG. 4

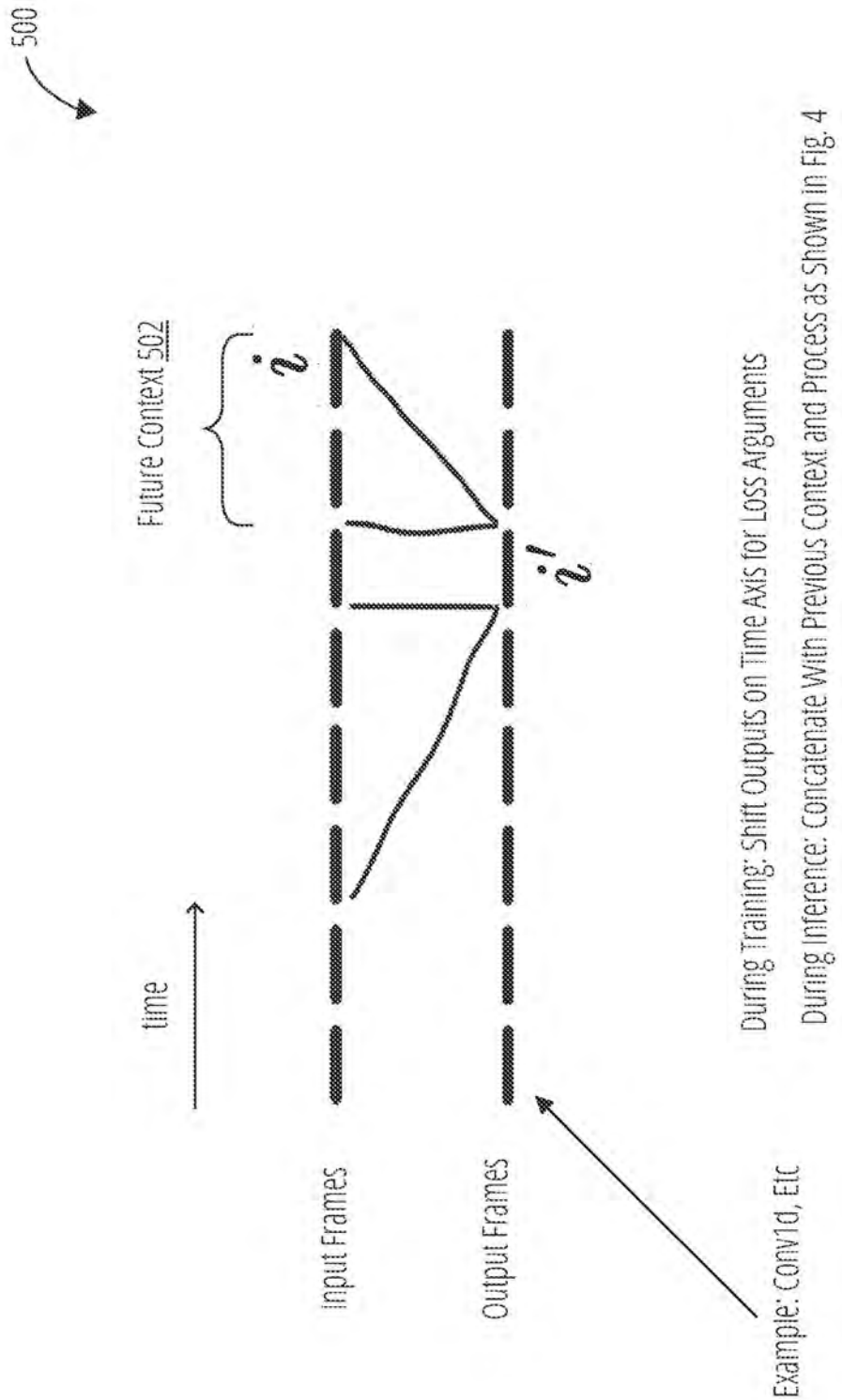

500
Future Context 502
i
i'
time
Input Frames
Output Frames
Example: Conv1d, Etc
During Training: Shift Outputs on Time Axis for Loss Arguments
During Inference: Concatenate With Previous Context and Process as Shown in Fig. 4

FIG. 5

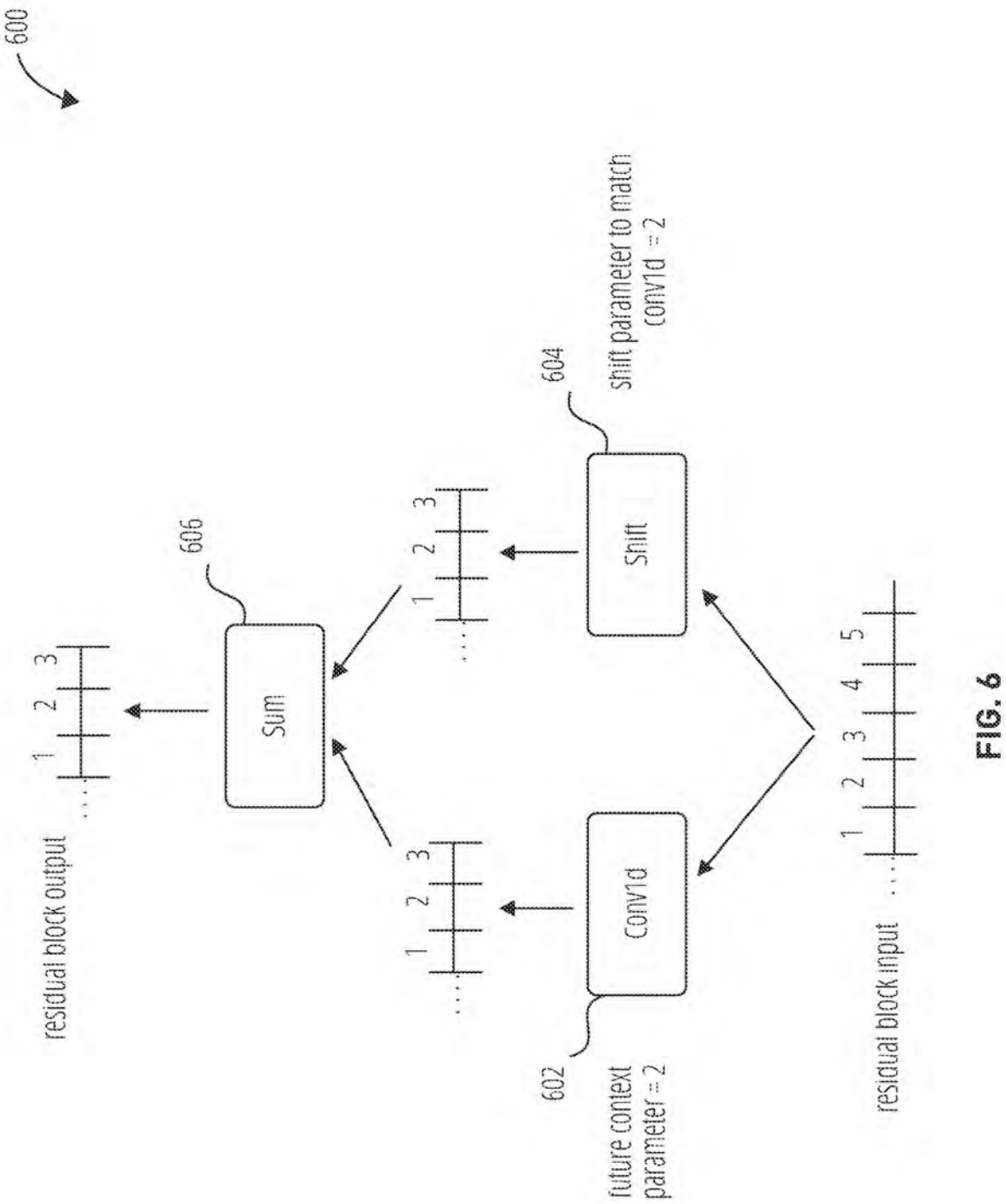

600
residual block input
1
2
3
4
5
602
future context parameter = 2
Conv1d
604
Shift parameter to match conv1d = 2
Shift
606
Sum
residual block output

FIG. 6

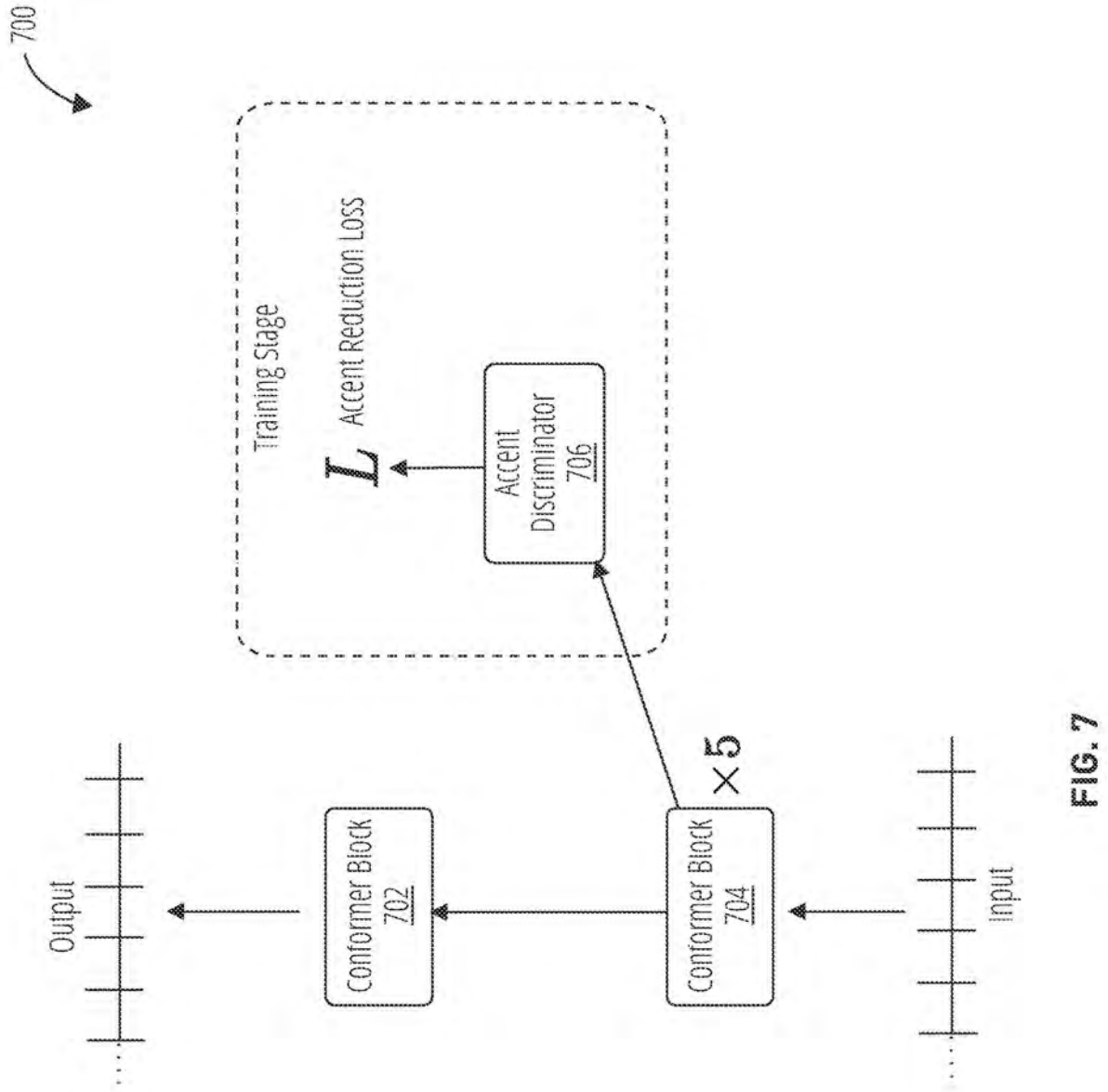

700
Training Stage
Accent Reduction Loss
L
Accent Discriminator 706
×5
Conformer Block 704
Conformer Block 702
Output
Input

FIG. 7

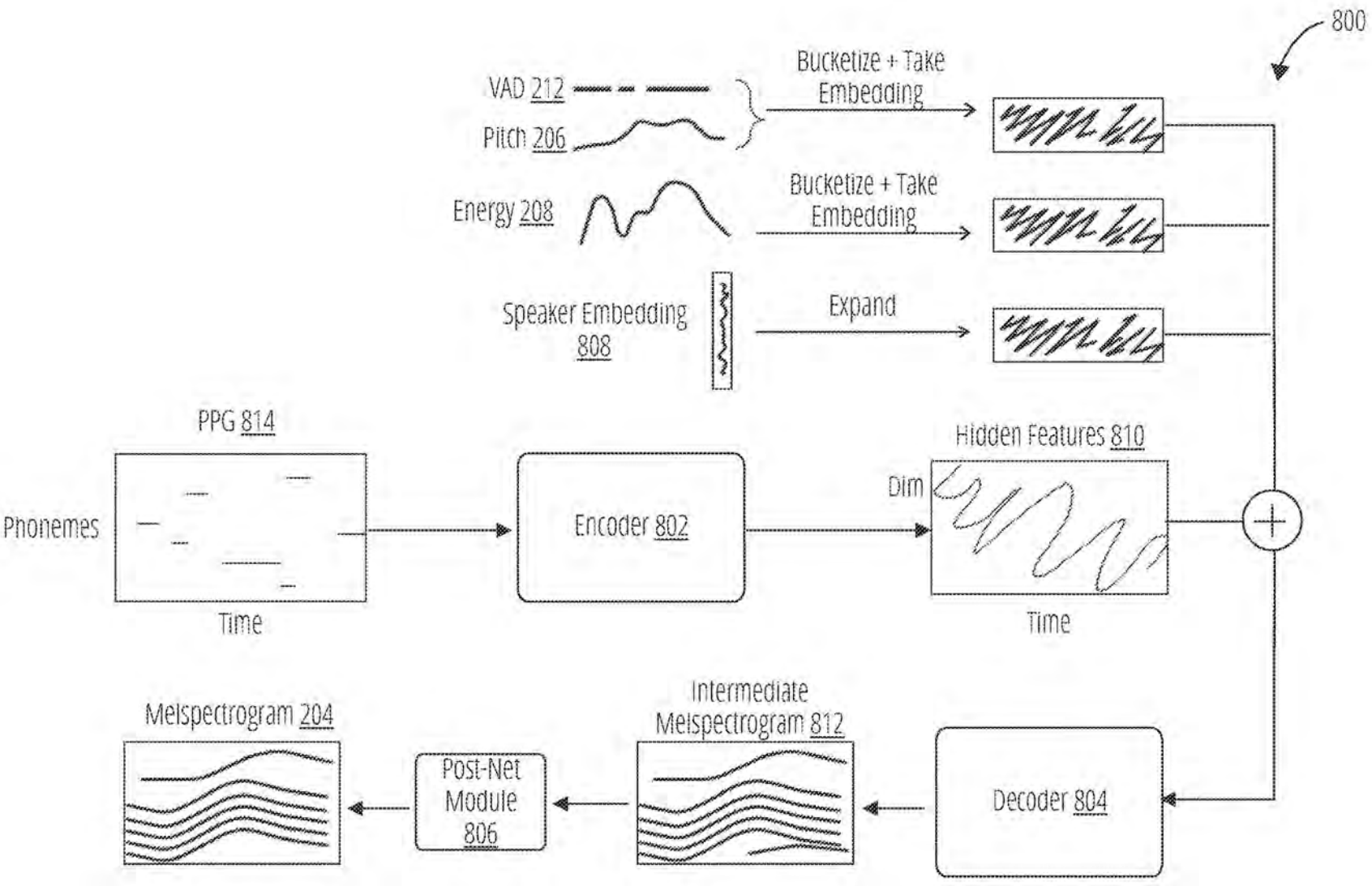

800
VAD 212
Pitch 206
Bucketize + Take Embedding
Energy 208
Bucketize + Take Embedding
Speaker Embedding 808
Expand
PPG 814
Phonemes
Time
Encoder 802
Hidden Features 810
Dim
Time
Intermediate Melspectrogram 812
Melspectrogram 204
Post-Net Module 806
Decoder 804

FIG. 8

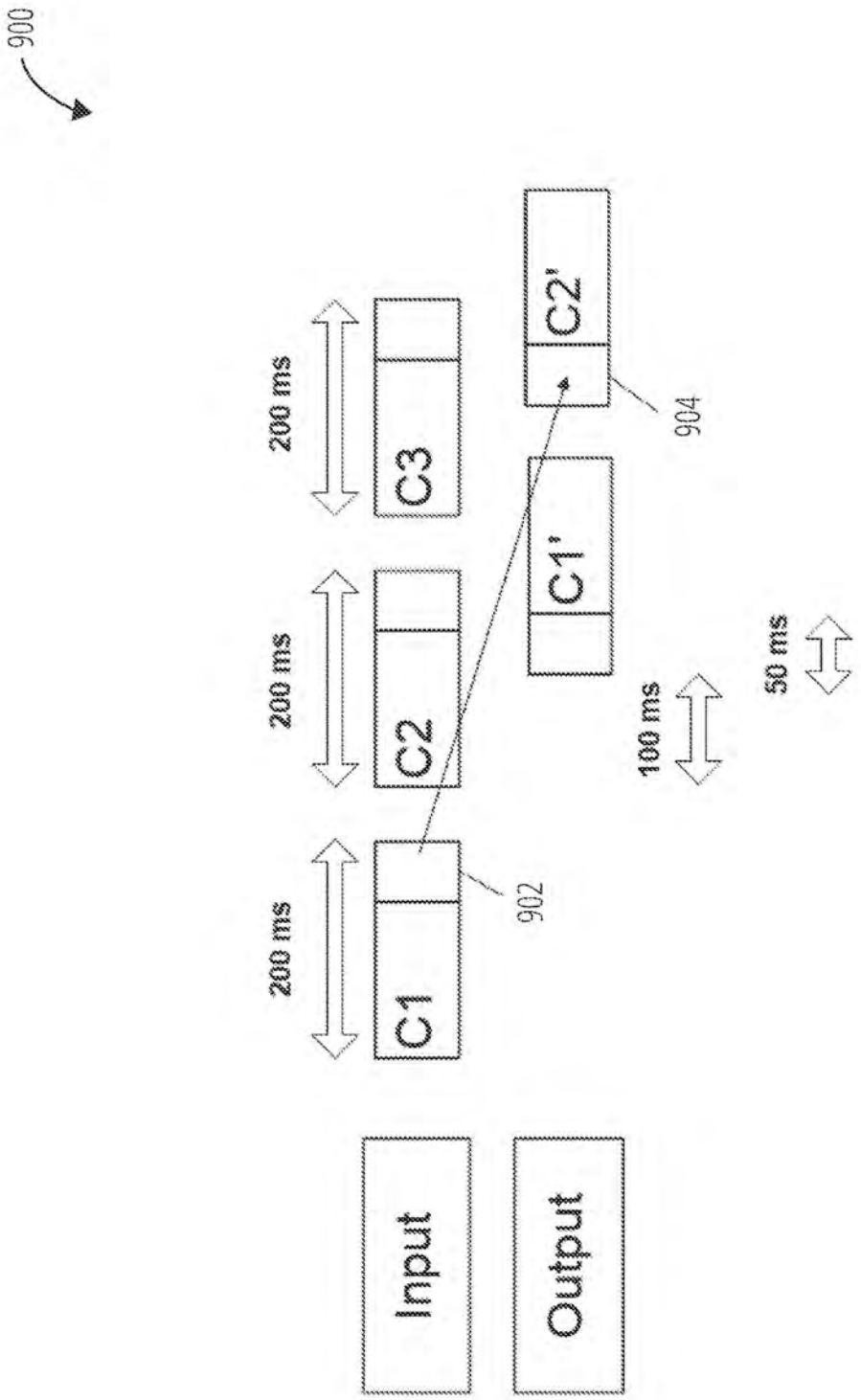

900
200 ms
200 ms
200 ms
C1
C2
C3
902
C1'
C2'
904
100 ms
50 ms
Input
Output

FIG. 9

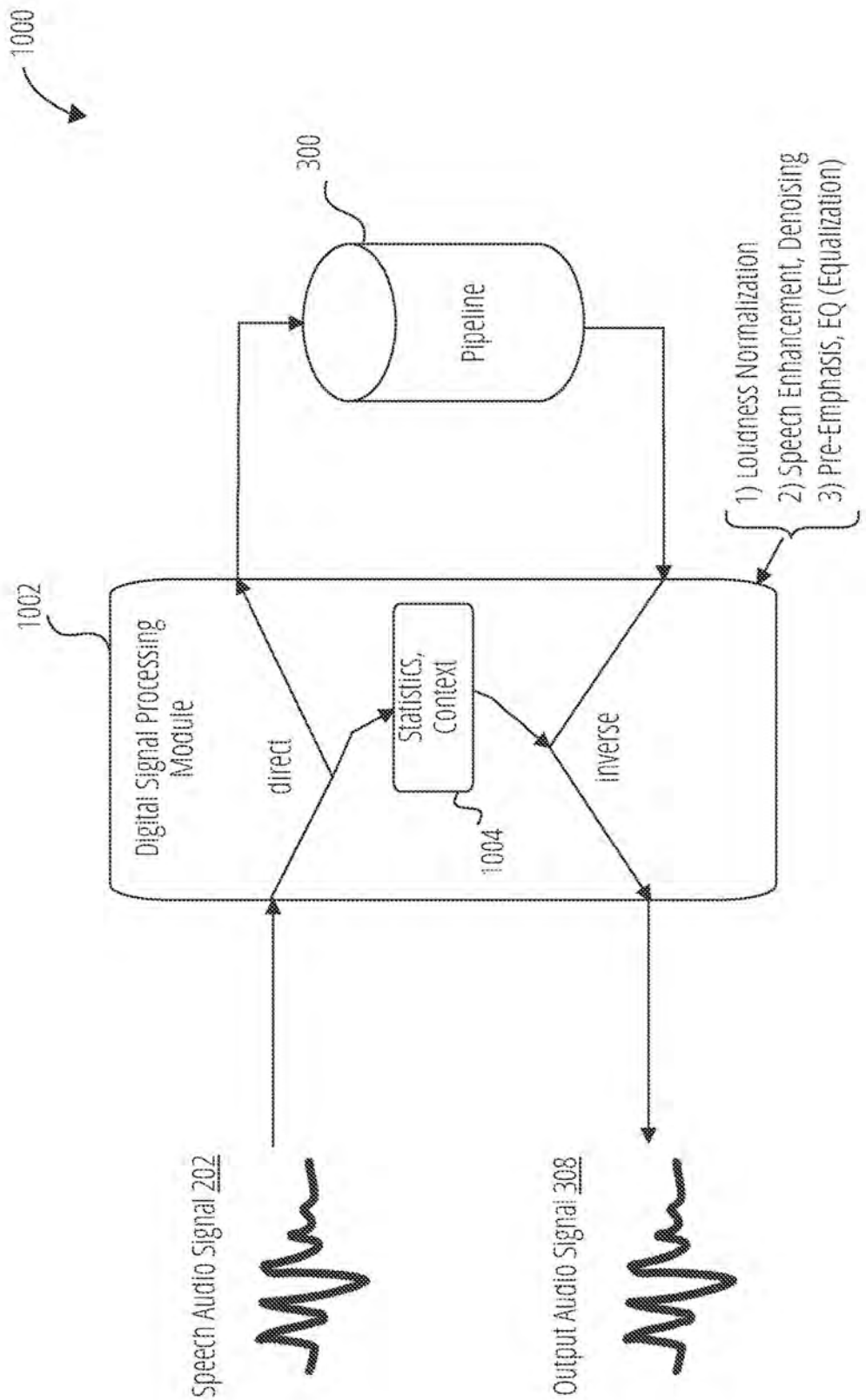

1000
300
Pipeline
1) Loudness Normalization
2) Speech Enhancement, Denoising
3) Pre-Emphasis, EQ (Equalization)
1002
Digital Signal Processing Module
direct
Statistics, Context
1004
inverse
Speech Audio Signal 202
Output Audio Signal 308

FIG. 10

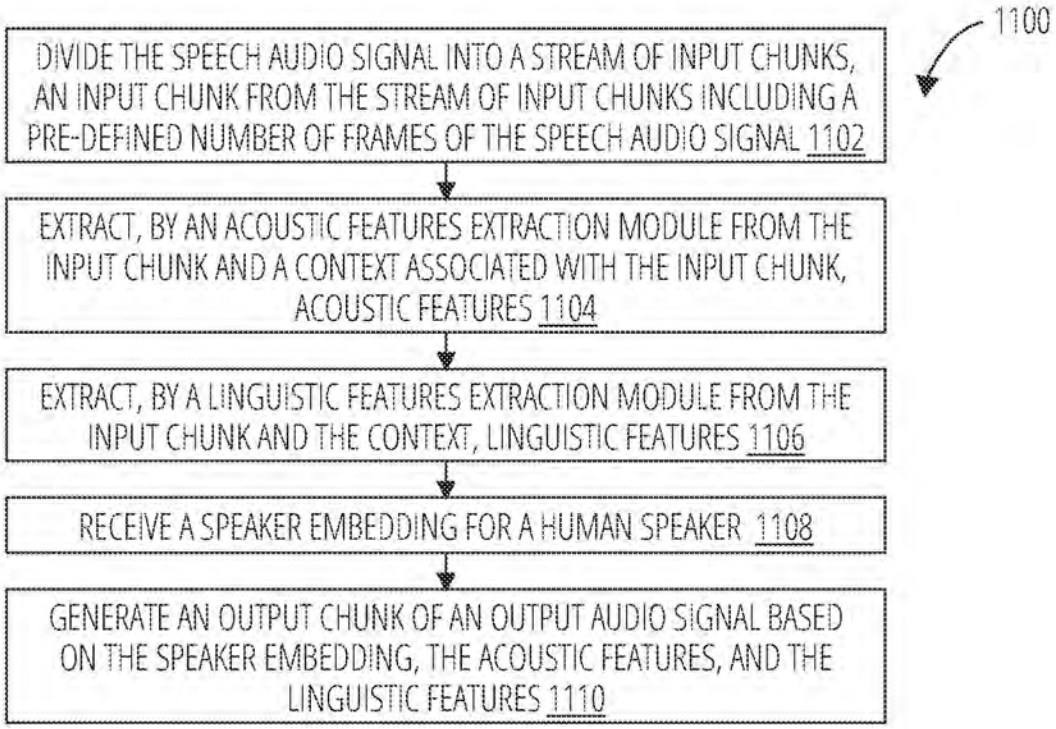

1100
DIVIDE THE SPEECH AUDIO SIGNAL INTO A STREAM OF INPUT CHUNKS, AN INPUT CHUNK FROM THE STREAM OF INPUT CHUNKS INCLUDING A PRE-DEFINED NUMBER OF FRAMES OF THE SPEECH AUDIO SIGNAL 1102
EXTRACT, BY AN ACOUSTIC FEATURES EXTRACTION MODULE FROM THE INPUT CHUNK AND A CONTEXT ASSOCIATED WITH THE INPUT CHUNK, ACOUSTIC FEATURES 1104
EXTRACT, BY A LINGUISTIC FEATURES EXTRACTION MODULE FROM THE INPUT CHUNK AND THE CONTEXT, LINGUISTIC FEATURES 1106
RECEIVE A SPEAKER EMBEDDING FOR A HUMAN SPEAKER 1108
GENERATE AN OUTPUT CHUNK OF AN OUTPUT AUDIO SIGNAL BASED ON THE SPEAKER EMBEDDING, THE ACOUSTIC FEATURES, AND THE LINGUISTIC FEATURES 1110

FIG. 11

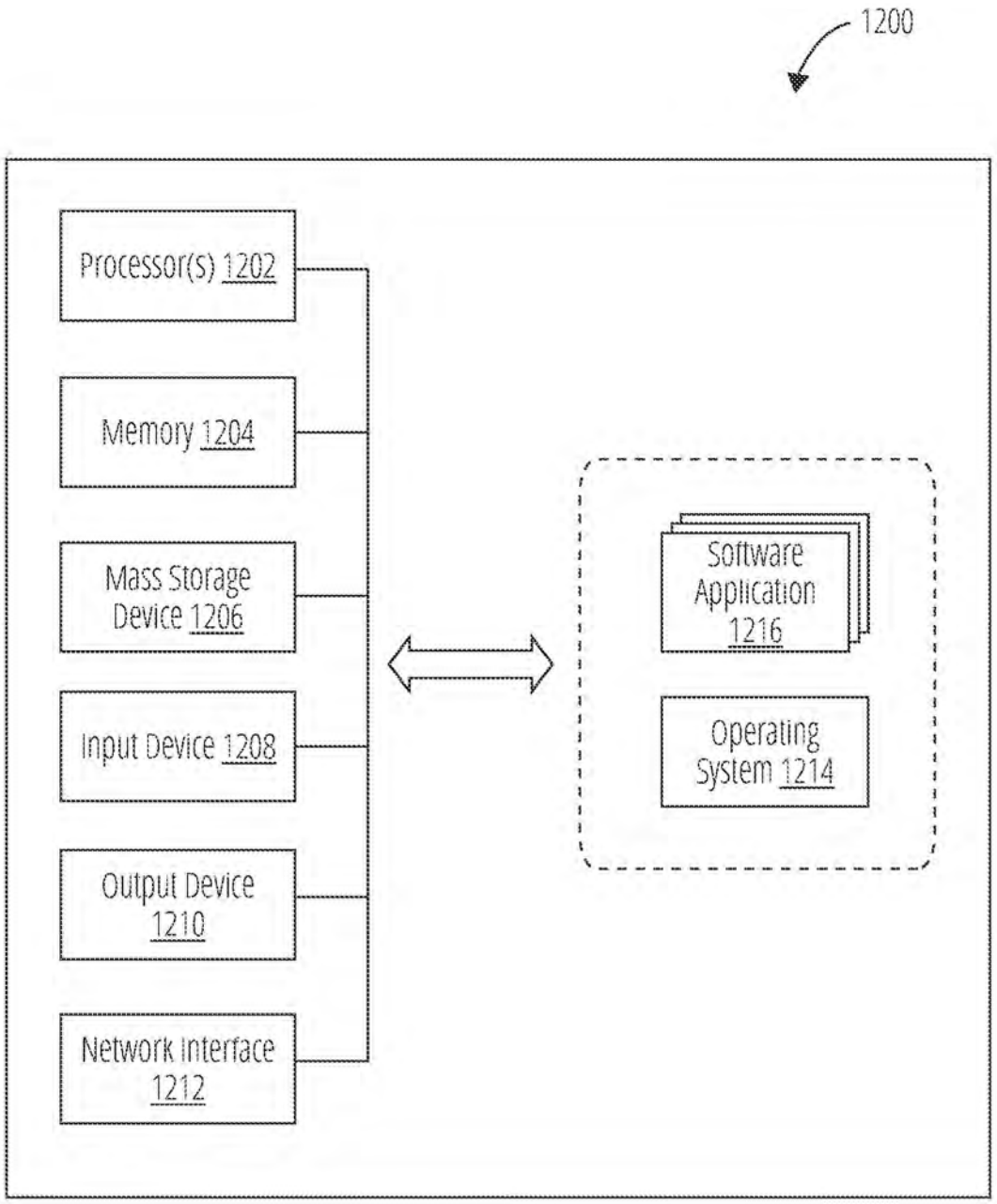

1200
Processor(s) 1202
Memory 1204
Mass Storage Device 1206
Input Device 1208
Output Device 1210
Network Interface 1212
Software Application 1216
Operating System 1214

FIG. 12

REAL TIME CORRECTION OF ACCENT IN SPEECH AUDIO SIGNALS

CROSS-REFERENCE TO RELATED APPLICATIONS

The present application claims priority of U.S. Provisional Patent Application No. 63/297,901 filed on Jan. 10, 2022, entitled "Real Time Correction of Accent in Speech Audio Signals," which is incorporated herein by reference in its entirety for all purposes.

TECHNICAL FIELD

This disclosure generally relates to audio processing. More particularly, this disclosure relates to systems and methods for real-time correction of accent in speech audio signals.

BACKGROUND

Audio conversations, such as audio chats, audio and video calls, and audio and video meetings are in wide use. One of the main problems encountered during an audio or video conversation is that speakers may possess strong accents that are difficult to understand by other participants. Existing solutions for correcting accent in audio signals are not very effective in real-time conversations.

SUMMARY

This summary is provided to introduce a selection of concepts in a simplified form that are further described in the Detailed Description below. This summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

According to one example embodiment of the present disclosure, a method for real-time correction of an accent in a speech audio signal is provided. The method can be implemented by a computing device and includes dividing the speech audio signal into a stream of input chunks. The input chunk from the stream of input chunks can include a pre-defined number of frames of the speech audio signal. The method may also include extracting, by an acoustic features extraction module from the input chunk and a context associated with the input chunk, acoustic features. The method may also include extracting, by a linguistic features extraction module from the input chunk and the context, linguistic features with a reduced accent or accent-agnostic linguistic features. The method may also include receiving a speaker embedding for a human speaker. The method may also include generating an output chunk of an output audio signal based on the speaker embedding, the acoustic features, and the linguistic features. The generation of the output chunk may include providing the speaker embedding, the acoustic features, and the linguistic features to a synthesis module to generate a melspectrogram with the reduced accent and, providing the melspectrogram to a vocoder to generate an output chunk of an output audio signal.

The context may include a pre-determined number of the frames belonging to chunks preceding the input chunk in the stream of input chunks. The speaker embedding can be pretrained based on audio data including a recorded speech of a target speaker having a further accent. Alternatively, the speaker embedding can be generated based on the speech audio signal in real-time.

The speech audio signal can be recorded from the voice of a user by the computing device via an acoustic sensor. A delay between the first timestamp and the second timestamp can be between 40 milliseconds and 300 milliseconds, where the first timestamp corresponds to the time when the chunk of the speech audio signal is recorded and the second timestamp corresponds to the time when the output chunk is generated.

The linguistic features may include one of the following: phonetic posteriorgrams with a standardized phonetic dictionary or phonetic posteriorgrams with a data-driven phonetic library. The linguistic features extraction module may include a neural network trained based on audio data to output the linguistic features. The neural network can be trained using a loss function to reduce, in the linguistic features, contributions due to a further accent present in the audio data.

The acoustic features can include a pitch of the speech audio signal, energy of the speech audio signal, and value of a voice activity detector, the voice activity detector indicating absence of human voice in the speech audio signal or presence of human voice in the speech audio signal.

The synthesis module may include an encoder, a decoder, and a post-net module designed to improve output of the decoder. Generating the melspectrogram may include processing the linguistic features by the encoder to generate hidden features, combining the hidden features, the acoustic features, and the speaker embeddings to generate further features, and processing the further features by the decoder and the post-net module to generate the melspectrogram.

The method may also include, prior to dividing the speech audio signal, processing the speech audio signal by a digital signal processing module to adjust one or more characteristics of the speech audio signal to improve extraction of the linguistic features and the acoustic features.

The method may also include, prior to dividing the speech audio signal, processing the speech audio signal by a digital signal processing module to adjust loudness of the speech audio signal from a first level to a second level. The method may also include, after generating the output chunk of the output audio signal, processing the output chunk by the digital signal processing module, to adjust loudness of the output audio signal to the first level.

According to another embodiment, a system for real-time correction of an accent in a speech audio signal is provided. The system may include at least one processor and a memory storing processor-executable codes, wherein the processor can be configured to implement the operations of the above-mentioned method for real-time correction of an accent in a speech audio signal.

According to yet another aspect of the disclosure, there is provided a non-transitory processor-readable medium, which stores processor-readable instructions. When the processor-readable instructions are executed by a processor, they cause the processor to implement the above-mentioned method for real-time correction of an accent in a speech audio signal.

Additional objects, advantages, and novel features will be set forth in part in the detailed description section of this disclosure, which follows, and in part will become apparent to those skilled in the art upon examination of this specification and the accompanying drawings or may be learned by production or operation of the example embodiments. The objects and advantages of the concepts may be realized and

attained by means of the methodologies, instrumentalities, and combinations particularly pointed out in the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments are illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements.

FIG. **1** shows an example environment, wherein a method for real-time correction of accent in speech audio signals can be practiced.

FIG. **2** is a schematic showing features that can be extracted from a speech audio signal, according to some example embodiments of the present disclosure.

FIG. **3** is a block diagram showing a pipeline for real-time correction of an accent in speech audio signals, according to an example embodiment.

FIG. **4** is a schematic showing details of dividing a speech audio signal into chunks for forming input data to modules of the pipeline, according to an example embodiment.

FIG. **5** is a schematic showing details of processing input frames during a training stage of submodules and modules of the pipeline, according to an example embodiment.

FIG. **6** is a schematic showing details of summation of a context in an example module **600**, according to an example embodiment.

FIG. **7** shows a part of an example neural network trained to generate linguistic accent-agnostic features, according to an example embedment.

FIG. **8** is a block diagram showing blocks of a synthesis module, according to an example embodiment.

FIG. **9** is a schematic showing details of streaming of a speech audio signal, according to some example embodiment.

FIG. **10** is a block diagram showing a digital signal processing module for use with a system for real-time correction of accent in speech audio signals, according to some example embodiments.

FIG. **11** is a flow chart showing a method **1100** for real-time correction of accent in speech audio signals, according to some example embodiments.

FIG. **12** is a high-level block diagram illustrating an example computer system, within which a set of instructions for causing the machine to perform any one or more of the methodologies discussed herein can be executed.

DETAILED DESCRIPTION

The following detailed description of embodiments includes references to the accompanying drawings, which form a part of the detailed description. Approaches described in this section are not prior art to the claims and are not admitted to be prior art by inclusion in this section. The drawings show illustrations in accordance with example embodiments. These example embodiments, which are also referred to herein as "examples," are described in enough detail to enable those skilled in the art to practice the present subject matter. The embodiments can be combined, other embodiments can be utilized, or structural, logical, and operational changes can be made without departing from the scope of what is claimed. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope is defined by the appended claims and their equivalents.

For purposes of this patent document, the terms "or" and "and" shall mean "and/or" unless stated otherwise or clearly intended otherwise by the context of their use. The term "a" shall mean "one or more" unless stated otherwise or where the use of "one or more" is clearly inappropriate. The terms "comprise," "comprising," "include," and "including" are interchangeable and not intended to be limiting. For example, the term "including" shall be interpreted to mean "including, but not limited to." The terms "can" and "may" shall mean "possibly be, but not limited to be."

This disclosure relates to methods and systems for real-time correction of accent in speech audio signals. Some embodiments of the present disclosure may be implemented in audio and video conversations to remove an accent in a speech audio signal captured from of a speaker uttering speech in a language that is not native to the speaker or a dialect of the same language that is different from the dialect spoken by other participants. Specifically, the speech audio signal can be analyzed in real time in chunks to extract acoustic features and linguistic features. The acoustic features and linguistic features can be then used to synthesize a melspectrogram lacking accent of the speaker. The melspectrogram can be used by a vocoder to generate an output audio signal lacking the accent.

In contrast to the existing solutions, embodiments of the present disclosure allow to reduce the delay between recording a chunk of the speech acoustic signal and outputting corresponding chunk of the output audio signal to 40-300 milliseconds.

FIG. **1** shows an example environment **100**, wherein a method for real-time correction of accent in speech audio signals can be practiced. It should be noted, however, that the environment **100** is just one example and is a simplified embodiment provided for illustrative purposes, and reasonable deviations of this embodiment are possible as will be evident to those skilled in the art.

As shown in FIG. **1**, environment **100** may include a user **102**, a user **104**, a computing device **106**, a computing device **110**, a network **108**, and a cloud-based computing resource **112** (also referred to as a computing cloud **112**).

The computing device **106** and computing device **110** each may include a sound sensor, memory, processor, communication unit, and output device. The memory may be configured to store processor-readable (machine-readable) instructions or codes, which when performed by the processor, cause the computing device **106** (or computing device **110**) to perform at least some steps of methods for real-time correction of accent in speech audio signals as described herein. The processor may perform floating point operations, complex operations, and other operations, including performing speech recognition and analysis based on ambient acoustic signals captured by sound sensor(s). The processors may include general purpose processors, video processors, audio processing systems, a central processing unit (CPU), a graphics processing unit (GPU), and so forth. The sound sensor(s) can include one or more microphones. The sound sensor(s) can be spaced a distance apart to allow the processor to perform a noise and/or echo reduction in received acoustic signals. The output device(s) may comprise one or more speaker(s), an earpiece of a headset, or a handset.

In various embodiments, the computing device **106** and computing device **110** can be configured to communicate with a network **108** such as the Internet, wide area network (WAN), local area network (LAN), cellular network, and so forth, to receive and send audio data.

The computing device **106** and computing device **110** can refer to a mobile device such as a mobile phone, smartphone, or tablet computer, a personal computer, laptop computer, netbook, set top box, television device, multimedia device,

personal digital assistant, game console, entertainment system, infotainment system, vehicle computer, or any other computing device. The computing device **106** can be communicatively connected to the computing device **110** and the computing cloud **112** via network **150**.

The network **108** can include any wired, wireless, or optical networks including, for example, the Internet, intranet, local area network (LAN), Personal Area Network (PAN), Wide Area Network (WAN), Virtual Private Network (VPN), cellular phone networks (e.g., Global System for Mobile (GSM) communications network, packet switching communications network, circuit switching communications network), Bluetooth™ radio, Ethernet network, an IEEE 602.11-based radio frequency network, a Frame Relay network, Internet Protocol (IP) communications network, or any other data communication network utilizing physical layers, link layer capability, or network layer to carry data packets, or any combinations of the above-listed data networks. In some embodiments, network **108** may include a corporate network, data center network, service provider network, mobile operator network, or any combinations thereof.

Computing cloud **112** can be shared by multiple users and be dynamically re-allocated based on demand. Computing cloud **112** can include one or more server farms and clusters including a collection of computer servers which can be co-located with network switches or routers.

According to one example embodiment, user **102** may communicate with user **104** through a voice call using a messenger or send voice messages via the messenger. The voice of the user **102** can be captured by the sound sensor of the computing device **106** to generate a speech audio signal. The user **102** may not be a native speaker of the language the user **102** speaks, so the speech audio signal may include an accent of the user **102**. The speech audio signal can be further modified to remove or reduce the accent of the user **102** in the speech audio signal.

In one embodiment, the modification of the speech audio signal can be carried out by a processor of computing device **106**. The modified speech audio signal can be sent, via the communication unit of the computing device **106**, to the computing device **110**. The computing device **110** may play back the modified speech audio signal via output device(s). Thus, user **104** may listen to the modified speech audio signal instead of the speech of the user **102**.

In other embodiments, the speech audio signal can be sent to the computing cloud **112**. In some embodiments, the speech audio signal can be sent to the computing cloud **112** using voice over internet protocol (VoIP). Computing cloud **112** can modify the speech audio signal to remove or correct the accent of the user **102** from the speech audio signal. Computing cloud **112** can send the modified speech audio signal to the computing device **110**.

FIG. **2** is a schematic showing features **216** that can be extracted from a speech audio signal **202**, according to some example embodiments of the present disclosure. The speech audio signal **202** may include waveforms **214**. The features **216** can be calculated per each time frame x. The features **216** may include acoustic features and linguistic features **210**. The acoustic features may include pitch **206** (or main frequency (F0)), energy **208** (signal amplitude), and voice activity detection (VAD) **212**. VAD **212** is a flag indicating the presence or absence of voice in the time frame.

Each of the features **216** is aligned with the others in time. Values of each feature are equidistant in time with respect to the values of the same feature obtained from the neighboring time frames. Accordingly, each of the features **216** is obtained from a chunk of speech audio signal **202** corresponding to an equal time period.

The melspectrogram **210** can be generated based on the acoustic features and linguistic features **210** as described herein.

FIG. **3** is a block diagram showing a pipeline **300** for real-time correction of an accent in speech audio signals, according to an example embodiment. The pipeline **300** may include an acoustic features extraction module **302**, a linguistic features extraction module **304**, a synthesis module **800**, and a vocoder **306**.

The acoustic features extraction module **302** may extract, from a time frame of the speech audio signal **202**, the acoustic features including pitch **206** (F0), energy **208**, and VAD **212**. These acoustic features can be obtained by algorithmic methods for signal processing or neural networks.

The linguistic features extraction module **304** may extract, from a time frame of the speech audio signal **202**, linguistic features **210**. In some embodiments, linguistic features **210** may include hidden features of an Automatic Speech Recognition (ASR) neural network with additional custom training and transformations or phonemes belonging to a phoneme set for a predetermined language. For example, the phoneme set may include ARPAbet phoneme set for English or classes (called pseudo-labels) of some clusterization algorithm over linguistic acoustic features of English speech data, like mel-spectrogram or hidden features of the ASR neural network. The phonemes can be obtained by a neural network trained to recognize and classify phonemes. In certain embodiments, the linguistic features **210** can be represented as Phonetic PosteriorGrams (PPGs). PPG can be defined as a distribution of the posterior probabilities of each phonetic class for each specific time frame of the speech audio signal **202**. Even though embodiments of the present disclosure are described as utilizing PPGs, the present technology can be practiced with any linguistic features.

The acoustic features and linguistic features can be provided to the synthesis module **800**. The synthesis module **800** may generate melspectrogram **204** corresponding to speech of the user **102** with removed or reduced accent. The melspectrogram **204** can be provided to the vocoder **306**. The vocoder **306** may generate output audio signal **308**.

FIG. **4** is a schematic **400** showing details of dividing a speech audio signal **202** into chunks for forming input data to modules of the pipeline **300**. The speech audio signal **202** can be provided to the modules (for example acoustic features extraction module **302**) as a stream **404** of chunks. Each chunk may include a pre-determined number of frames. Each of the frames is a portion of the speech audio signal **202** of a predetermined time interval size. In some embodiments, the length of each of the frame can be, for example, 11.6 milliseconds (ms).

An input to module **302** may include a chunk **408** of frames concatenated with a context **402**. The context **402** may include a pre-determined number of frames of the speech audio signal **202** preceding the chunk **408**. Context **402** can be stored in a cache of module **302** and continuously updated. Thus, at each state of real time, input of the modules can include the chunk **408** ended at the previous state of real time and the context **402** corresponding to the chunk.

The output of module **302** is stream **440**. The stream **404** may include chunks of one of acoustic features (pitch **206** (F0), energy **208**, or VAD **212**. Output chunk **410** can be formed by cutting, from the output stream **406**, a chunk that

ends at an effective state of real time in module **302**. The context **402** can be extended by chunk **408**. The first chunk in the context **402** can be removed. Thus, the modified context **402** can be used for processing the next chunk from the stream **404**. The output chunk **410** can be provided to the synthesis module **800** (shown in FIG. **3**). Similarly, module **304** (shown in FIG. **3**) may produce a stream of chunks of linguistic features. The chunks of linguistic features can also be provided to the synthesis module **800**.

Overall, input of the synthesis module **800** includes a stream of chunks of linguistic features **225** (PPGs), a stream of chunks of values of pitch **206** (F0), a stream of chunks of values of energy **208**, and a stream of chunks of values of VAD **212**, all the streams being aligned with each other. The output of the synthesis module **800** module and, correspondently, the input of the vocoder **306**, includes a stream of chunks of melspectrogram **204**. Similarly, to module **302**, each of the modules **304**, **800**, and **306** may have a cache to store a context including predetermined number of previous frames of the corresponding features. The above architecture of streaming chunks to every one of modules **302**, **304**, **800**, and **306** can be applied recursively to internal submodules of these modules, such as neural network blocks and layers.

In further embodiments, the context can be also cached for submodules of the modules **302**, **304**, **800**, and **306**. For example, acoustic features extraction module **302** may include one or more of the following submodules: 1D convolution layer (Conv1d), attention layer, and variance predictors. Each of the submodules may include cache for storing the context of output of corresponding preceding submodule in the acoustic features extraction module **302**. The preceding submodule may output a stream of chunks of internal features corresponding to stream **404**. The input to the next submodule in module **302** may include the last output chunk produced by the preceding submodule and the context including a predetermined number of previous frames of chunks produced by the preceding submodule.

Caching context for inner submodules of modules **302**, **304**, **800**, and **306** (outer modules) may allow to achieve same output quality for modules **302**, **304**, **800**, and **306** between training stage and streaming (inference) stage because a future context of an outer module originates from future contexts of inner submodules. Every layer in sequential part of a neural network that implements one of the modules **302**, **304**, **800**, and **306** can be part of the future context. The parts of the future context can be summed up to receive total a future context of the outer module. The total future context of the outer module can be split into the outer part (regulated with cache of the outer module) and inner part (regulated with inner submodules' caches). In some embodiments, only inner future context can be used in streaming. In other embodiments, partially inner further context and partially outer future context can be used in streaming.

FIG. **5** is a schematic **500** showing details of processing input frames during the training stage of submodules and modules of the pipeline **300**, according to an example embedment. FIG. **5** shows input frames and output frames for a minimal example of a neural network layer, which produces output shifted on the time axis parametrized by future context (also referred to as a shift). For example, the neural network layer may include conv1d, attention layer, conformer and other layers. During training, the output frames i' can be shifted for calculation of Loss sensitive to time location, which can teach the model (layer) to produce shifted output by the parameter of the future context. On inference stage (streaming) the input frames i can be concatenated with previous context, divided in chucks and processed as described in FIG. **4**.

FIG. **6** is a schematic showing details of summation of a context in an example module **600**, according to an example embedment. The module **600** may include parallel blocks, Conv1D **602** and Shift **604**. The Conv1D **602** may use input frames 1, 2, 3 as context for input frames 4 and 5. To obtain the total future context for module **600**, the input frames 1, 2, 3, 4, 5 can be shifted by shift **604** by 2 frames and summed with context output from the Conv1D **602** in block **606**.

In some embodiments, the future context can be determined as a maximum of sums of context in any sequential path within the module, submodule or neural network. For example, the residual block module has a residual connection of a convolution layer with two sequential operations, where the first operation is convolution with future context x, and the second operation is residual summation of inputs to convolution layer to output. There are two sequential paths from inputs to outputs in such a module, the first path: inputs→conv→add→output, and the second path: inputs→add→output. If the sequential path with the maximum sum of future contexts is the first path, then the total future context equals x.

Technical Details

1) In some embodiments, the speech audio signal **202** can include a continuous stream of 16-bit samples with a frequency of 22050 kHz. The speech audio signal **202** can be split into overlapping windows with a step of 256 samples (~11.6 ms). Accordingly, one frame corresponds to 256 samples.

The acoustic features and linguistic features can be extracted and calculated such that the centers of the windows that correspond to each feature coincide. Thus, the centers can point to the same moment in time to satisfy the condition of alignment of the features. Accordingly, when the synthesis module **800** processes the input, the number of frames from each feature is the same. The number of frames in the output melspectrogram **204** may also coincide with the number of frames from each feature. Consequently, the number of samples of the speech audio signal **202** (input signal) can be equal to the number of samples of output audio signal **308**.

2) FIG. **7** shows a part of an example neural network **700** trained to generate accent-agnostic PPGs. The neural network **700** may include Conformer blocks **704** and Conformer block **702**. The Conformer is a convolution-augmented transformer for speech recognition. Each of the Conformer blocks can be implemented with restrictions on attention and the convolutional layer on visible future frames and previous frames. Neural network **700** trained to generate PPGs may also include a linear head for predicting phonemes.

The neural network **700** can be trained using an accent discriminator **706** and supervised information for accent. During training, additional feed-forward network (FFN) can be used between transformer blocks of neural network **700**. Output features from FFN and supervised label on accent can be utilized for additional accent reduction loss L. Training with the additional accent reduction loss may reduce leak of accent through the recognition model. In example of FIG. **7**, output of the fifth Conformer block **704** can be utilized to produce additional features by simple feed-forward network (for example linear-Rectified Linear Unit (ReLU)-linear). These features can be utilized for accent reduction loss based on data labels indicating which

accent is used on every utterance. Use of the accent reduction loss during training may help to produce accent agnostic features). "Cross-entropy classification loss with reversal gradients module" for the accent reduction loss.

During inference, an output (target) accent can be selected from accents available on training stage. During the training stage, datasets of different voices and accents can be used. Any of the datasets can be validated for appropriate sound quality and then used for output target voice and accent.

3) Extraction of acoustic features (pitch **206** (F0), energy **208**, or VAD **212**) can be performed by algorithmic methods using sound processing tools or by trained neural networks. The following algorithmic methods and utilities may be applied:

- Energy **208**: Short-time Fourier transform (STFT) followed by a summation over all frequency bins and applying a logarithm to the result of the summation.
- Pitch **206** (F0) and VAD **212**: values of F0 and voiced/unvoiced intervals can be obtained using the pyWORLD script. pyWORLD is a free software for high-quality speech analysis, manipulation and synthesis. The pyWORLD can estimate fundamental frequency (F0), aperiodicity, and spectral envelope. The values of F0 can be interpolated to unvoiced intervals. Then, the logarithm can be applied to resulting F0.
- Energy **208** and Pitch **206** (F0) can also be normalized globally using average variance of corresponding values obtained from voice signals recorded from multiple speakers.

FIG. **8** is a block diagram showing blocks of the synthesis module **800**, according to an example embodiment. The synthesis module **800** may include an encoder **802** and decoder **804**. In some embodiments, encoder **802** and decoder **804** can be implemented as neural networks. Specifically, both encoder **802** and decoder **804** can be based on lightweight convolution blocks. A convolutional layer (Conv1d-Groupnorm-GELU) acting as relative positional embedding can be applied to input of the encoder **802**. In some embodiments, relative positional embedding is added to the input inside the encoder **802**. A further convolutional layer (Conv1d-Groupnorm-GELU) acting as further relative positional embedding can be applied to input of the decoder **804**. In some embodiments, the further relative positional embedding is added to the input of the decoder **804**.

The input of the encoder **802** are linguistic features **210**. The output of the encoder **802** has hidden features **810**. The speaker embedding **808** of a target speaker and embeddings of the discretized values of energy **208** and pitch **206** (f0) are further added to the output of the encoder **802** to form input for the decoder **804**. If VAD **212**=False, a separate embedding is used instead of embedding of pitch **206** (F0). Speaker embedding **808** can be a calculated feature in the form of a dense multi-dimensional vector. Speaker embeddings **808** may include necessary information on target speakers' voice style not related to the accent of the target speaker.

In various embodiments, speaker embedding **808** can be trained fixed, pre-trained fixed, or extracted via a pre-trained model from speech audio signal **202** in real-time. For example, the speaker embedding **808** can be trained or extracted using pre-trained algorithms in such a way that the voice acoustic features corresponding to the speaker embedding **808** match voice acoustic features of the target speaker. In these embodiments, the speaker embedding **808** can be pretrained based on audio data including recorded speech of the target speaker. The user **102** may be provided with an option to select speaker embedding **808** form a list of pretrained speaker embeddings corresponding to different speakers.

In other embodiments, the voice speaker embedding **808** can be generated in real-time based on speech audio signal **202** being recorded from the voice of the user **102**. In these embodiments, a caching scheme similar to the caching scheme described in FIG. **4** can be used to extract the speaker embedding **808** from speech audio signal **202** in real-time. The speaker embedding **808** can be used later to produce output audio signal **308** having voice acoustic features of the user **102**.

In yet other embodiments, the speaker embedding **808** can be pre-generated based on previously recorded speech signals of the user **102** and stored in memory of the computing device **106** or computing cloud **112**. In these embodiments, the speaker embedding **808** can be retrieved from the memory computing device **106** or computing cloud **112** to avoid recomputing the speaker embedding **808** in real-time.

The output of decoder **804** is an intermediate melspectrogram **812**. The intermediate melspectrogram **812** is used in a post-net module **806** to output melspectrogram **210**. The post-net module **806** can be implemented as a small convolutional network. In some embodiments, the post-net module **806** can be similar to a post-net used in Tacotron 2. The values of VAD **212**, Energy **208**, and pitch **206** can be the same as acoustic features extracted by acoustic features extraction module **302** (shown in FIG. **3**) or predicted by separate modules of the synthesis module **800**. All blocks of encoder **802** and decoder **804**, as well as predictors, the relative positional encoding layer, and the post-net module **806** can be implemented with a limited future context.

The output of the post-net module **806** is provided to vocoder **306**. In some embodiments, the vocoder **306** can correspond to the HiFi-GAN v2 or LPCNet vocoder without changes. The vocoder parameters may correspond to the synthesis of the audio signal for the frames of the melspectrogram.

Data

- For PPG. Medium-quality voice data of various accents with the presence of texts are available in datasets of LibriSpeech and CommonVoice. The texts can be normalized and processed to obtain phoneme sequences according to ARPAbet phoneme set. Next, the procedure of alignment (align) of phonemes in time can be performed using the Montreal-Forced-Aligner utility. The image of texts can be processed by grapheme-to-phoneme (g2p) to obtain phonemes. Then, the phonemes can be processed and aligned together with audio signal.
- For the vocoder. The VCTK dataset is used for pre-training and, similarly to the data for synthesis, pure data from the same speakers that were not used to train the synthesis model. These data are resynthesized to melspectrograms. The melspectrograms can then be used together with the original pure audio to retrain the vocoder.

Training

In some embodiments, the PPG model is trained in two stages: pre-training and additional training. A set of augmentations, such as noise and SpecAugment can be used in both stages.

Pre-training of the PPG model can be performed in an unsupervised manner and using clustering. Specifically, Mel-frequency cepstral coefficients (Mfcc) or hidden fea-

tures of large ASR neural networks can be algorithmically divided into clusters using k-means. Each frame can be assigned to a specific cluster (by a number). Pre-training includes training the PPG model with a classification head to obtain the number of the cluster for a frame. The last hidden layer of features of the PPG model can be clustered (like mfcc) and used for training an improved PPG model. This procedure can be applied iteratively.

Additional training of the PPG model is carried out on connectionist temporal classification loss (recognition task) by phoneme sequence and cross-entropy loss (classification task) by phoneme prediction in each frame. To do this, two appropriate heads can be used on top of the encoder in the PPG model. As described in FIG. **7**, additional training of PPG model can be performed using loss for accent reduction. In these embodiments, the training can be performed by providing output of one of intermediate blocks of the PPG model to an accent discriminator with reversal gradient using accent classification loss function, which may play the role of accent loss function (see blocks **704** and **706** in FIG. **7**).

The synthesis model can be trained on predictions of acoustic values in predictors of values of the output mel-spectrogram after the decoder and after the post-network. The predictors may include mean squared error (mse) loss according to energy and f0 predictions and binary cross entropy loss according to VAD prediction. For the synthesis model, output speaker embeddings can be trained as parameters which lead to a fixed number of available output speaker embeddings. In other embodiments, the output speaker embeddings can be obtained as hidden features of a pre-trained speaker classification model applied in streaming manner to input speech data in order to perceive input speaker voice.

Vocoder can be trained in two stages: training on a large multiple speaker dataset and additional training on resynthesis with the help of the already trained part of the pipeline **300**. The optimization methods (training methods) can be combined to train described models jointly. During joint training, a single audio sample can be used for every loss function calculation and every parameter update.

Streaming

FIG. **9** is a schematic showing details of streaming of a speech audio signal, according to some example embodiment.

The speech audio signal can be processed in chunks. A chunk may correspond to a certain window, typically, 3~10 frames=3*256~10*256 samples=35~116 ms. Each module in pipeline **300** processes the chunk and outputs a result corresponding to the size of the input chunk.

In accordance with architectures of modules (feature extraction, synthesis, vocoding), an appropriate number of frames/samples can be cut off (modularly or at the output) to obtain a high-quality result with a low latency. The number of frames/samples can be defined as the total number of frames/samples from the front of the signal.

The streaming delay can be defined as the time difference between the original moment of speech and the output corresponding to the original one. The streaming delay includes the following components:

- Architectural delay. This delay is embedded in the indentation to account for a larger future context and thereby improving the processing quality of each module in the pipeline.
- The size of the chunk. The size of the chunk affects time for waiting for all the data before processing because the modules cannot output the result until the data are obtained by the modules.
- Processing time. The processing time is a time within which the chunk is completely processed by modules in the pipeline **300**. The processing time needed to be adjusted to ensure a stable conversion of input chunks into output chunks.

In the example of FIG. **9**, the size of the chunk is 200 ms, maximum processing time is 100 ms, and the architecture delay is 50 ms. The total delay is 350 ms. The chunks C1, C2, C3, . . . are fed into the pipeline **300** in real time. Each output chunk C1', C2', . . . correspond to only one of the input chunk C1, C2, C3, ... portion **904** of output audio signal **308** corresponds to portion **902** of input speech audio signal **202**. Portion **902** and portion **904** correspond to the same moment of the speech of the user **102**. Overall, according to experiments conducted by the inventors, the methods of the present disclosure allow to achieve the total delay of 40-300 ms.

FIG. **10** is a block diagram **1000** showing a digital signal processing module **1002** for use with pipeline **300** for real-time correction of accent in speech audio signals, according to some example embodiments. The digital signal processing module **1002** can be used for enhancement of speech audio signal **202** and output audio signal **308**. The digital signal processing module **1002** may include submodule **1004** for collecting and storing statistics and context during processing speech audio signal **202**. The processed speech audio signal **202** can be further provided to pipeline **300** for correction of accent. The output of the pipeline **300** can be processed by digital signal processing module **1002** to obtain an output audio signal **308**. During processing the output of pipeline **300** the digital signal processing module **1002** can utilize statistics and context collected by submodule **1004** to restore some characteristics removed from the speech audio signal.

In some embodiments, digital signal processing module **1002** may process speech audio signal **202** to remove or attenuate noise, cancel echo, and remove other artifacts. Digital signal processing module **1002** may also perform normalization of loudness of the signal, equalizing the signal, applying a pre-emphasis or de-emphasis to the signal, and enhancing a speech in the signal. In certain embodiments, digital signal processing module **1002** can be integrated in one of the modules of the pipeline **300** as a beginning submodule or inserted between any two modules of the pipeline **300**. In these embodiments, digital signal processing module **1002** can be trained with corresponding losses to imitate digital signal processing algorithms.

In some embodiments, digital signal processing module **1002** can be used to control loudness of output audio signal **308**. For example, digital signal processing module **1002** may auto-gain loudness of speech audio signal **202** before pipeline **300** processing and then, based on a user setting, restore or not to restore level of loudness of output audio signal **308** to corresponding level of loudness of speech audio signal **202**.

FIG. **11** is a flow chart showing a method **1100** for real-time correction of accent in speech audio signals, according to some example embodiments. In some embodiments, the operations of method **1100** may be combined, performed in parallel, or performed in a different order. The method **1100** may also include additional or fewer operations than those illustrated. The method **1100** may be performed by processing logic that comprises hardware (e.g.,

decision making logic, dedicated logic, programmable logic, and microcode), software (such as software run on a general-purpose computer system or a dedicated machine), or a combination of both.

In block **1102**, method **1100** may divide the speech audio signal into a stream of input chunks, an input chunk from the stream of input chunks including a pre-defined number of frames of the speech audio signal. The speech audio signal can be recorded, via an acoustic sensor, from the voice of a user by a computing device implementing method **1100**.

In block **1104**, method **1100** may extract, by an acoustic features extraction module from the input chunk and a context associated with the input chunk, acoustic features. The context may include a pre-determined number of the frames belonging to chunks preceding the input chunk in the stream of input chunks. The acoustic features may include a pitch of the speech audio signal, an energy of the speech audio signal, and a value of a voice activity detector. The voice activity detector may indicate absence of a human voice in the speech audio signal or presence of the human voice in the speech audio signal.

In block **1106**, method **1100** may extract, by a linguistic features extraction module from the input chunk and the context, linguistic features with a reduced accent or accent-agnostic linguistic features. The linguistic features extraction module may include a neural network trained based on audio data to output the linguistic features, neural network being trained using a loss function to reduce, in the linguistic features, contributions due to a further accent present in the audio data. The linguistic features may include one of the following: phonetic posteriorgrams or phonetic posteriorgrams with a data-driven phonetic library.

In block **1108**, method **1100** may receive a speaker embedding for a human speaker. The speaker embedding can be pretrained based on audio data including a recorded speech of a target speaker having a further accent. Alternatively, the speaker embedding can be generated based on the speech audio signal in real-time.

In block **1110**, method **1100** may generate an output chunk of an output audio signal based on the speaker embedding, the acoustic features, and the linguistic features. For example, method **1100** may provide the speaker embedding, the acoustic features, and the linguistic features to a synthesis module to generate a melspectrogram with the reduced accent. The synthesis module may include an encoder, a decoder, and a post-net module designed to improve the output of the decoder. Generating the melspectrogram may include processing the linguistic features by the encoder to generate hidden features; combining the hidden features, the acoustic features, and the speaker embeddings to generate further features; processing the further features by the decoder and the post-net module to generate the melspectrogram.

After generating the melspectrogram, method **1100** may provide the melspectrogram to a vocoder to generate an output chunk of an output audio signal. A delay between the first timestamp corresponding to the time when the chunk of the speech audio signal is recorded and the second timestamp corresponding to the time when the output chunk is generated can be between 40 ms and 300 ms.

The acoustic features can be split into a stream of acoustic features chunks corresponding to the chunks in the stream of input chunks. The linguistic features can be split into a stream of linguistic features chunks corresponding to the chunks in the stream of input chunks. The melspectrogram can be split into a stream of melspectrogram chunks corresponding to the chunks in the stream of input chunks. A melspectrogram chunk of the stream of melspectrogram chunks is generated based on the following:

- an acoustic features chunk of the stream of acoustic features chunks and acoustic features context including the pre-determined number of acoustic features frames belonging to acoustic features chunks preceding the acoustic features chunk in the stream of acoustic features chunks; and
- a linguistic features chunk of the stream of melspectrogram chunks and linguistic features context including the pre-determined number of linguistic features frames belonging to linguistic features chunks preceding the linguistic features chunk in the stream of acoustic features chunks.

The method **1100** may include, prior to dividing the speech audio signal, processing the speech audio signal by a digital signal processing module to adjust one or more characteristics of the speech audio signal to improve extraction of the linguistic features and the acoustic features.

The method **1100** may include, prior to dividing the speech audio signal, processing the speech audio signal by a digital signal processing module to adjust a loudness of the speech audio signal from a first level to a second level. Method **1100** may include, after generating the output chunk of the output audio signal, processing the output chunk by the digital signal processing module, to adjust the loudness of the output audio signal to the first level.

FIG. **12** is a high-level block diagram illustrating an example computer system **1200**, within which a set of instructions for causing the machine to perform any one or more of the methodologies discussed herein can be executed. The computer system **1200** may include, refer to, or be an integral part of, one or more of a variety of types of devices, such as a general-purpose computer, a desktop computer, a laptop computer, a tablet computer, a netbook, a mobile phone, a smartphone, a personal digital computer, a smart television device, and a server, among others. In some embodiments, the computer system **1200** is an example of computing devices **106**, computing device **110**, and computing cloud **112** shown in FIG. **1**. Notably, FIG. **12** illustrates just one example of the computer system **1200** and, in some embodiments, the computer system **1200** may have fewer elements/modules than shown in FIG. **12** or more elements/modules than shown in FIG. **12**.

The computer system **1200** may include one or more processor(s) **1202**, a memory **1204**, one or more mass storage devices **1206**, one or more input devices **1208**, one or more output devices **1210**, and a network interface **1212**. The processor(s) **1202** are, in some examples, configured to implement functionality and/or process instructions for execution within the computer system **1200**. For example, the processor(s) **1202** may process instructions stored in the memory **1204** and/or instructions stored on the mass storage devices **1206**. Such instructions may include components of an operating system **1214** or software applications **1216**. The computer system **1200** may also include one or more additional components not shown in FIG. **12**, such as a body, a power supply, a power supply, a global positioning system (GPS) receiver, and so forth.

Memory **1204**, according to one example, is configured to store information within the computer system **1200** during operation. The memory **1204**, in some example embodiments, may refer to a non-transitory computer-readable storage medium or a computer-readable storage device. In some examples, memory **1204** is a temporary memory, meaning that a primary purpose of the memory **1204** may not be long-term storage. Memory **1204** may also refer to a

volatile memory, meaning that memory **1204** does not maintain stored contents when the memory **1204** is not receiving power. Examples of volatile memories include random access memories (RAM), dynamic random access memories (DRAM), static random access memories (SRAM), and other forms of volatile memories known in the art. In some examples, memory **1204** is used to store program instructions for execution by the processor(s) **1202**. The memory **1204**, in one example, is used by software (e.g., the operating system **1214** or the software applications **1216**). Generally, the software applications **1216** refer to software Applications suitable for implementing at least some operations of the methods for real-time correction of accent in speech audio signals as described herein.

The mass storage devices **1206** may include one or more transitory or non-transitory computer-readable storage media and/or computer-readable storage devices. In some embodiments, the mass storage devices **1206** may be configured to store greater amounts of information than the memory **1204**. The mass storage devices **1206** may further be configured for long-term storage of information. In some examples, the mass storage devices **1206** include non-volatile storage elements. Examples of such non-volatile storage elements include magnetic hard discs, optical discs, solid-state discs, flash memories, forms of electrically programmable memories (EPROM) or electrically erasable and programmable memories, and other forms of non-volatile memories known in the art.

Input devices **1208**, in some examples, may be configured to receive input from a user through tactile, audio, video, or biometric channels. Examples of the input devices **1208** may include a keyboard, a keypad, a mouse, a trackball, a touchscreen, a touchpad, a microphone, one or more video cameras, image sensors, fingerprint sensors, or any other device capable of detecting an input from a user or other source, and relaying the input to the computer system **1200**, or components thereof.

The output devices **1210**, in some examples, may be configured to provide output to a user through visual or auditory channels. The output devices **1210** may include a video graphics adapter card, a liquid crystal display (LCD) monitor, a light emitting diode (LED) monitor, an organic LED monitor, a sound card, a speaker, a lighting device, a LED, a projector, or any other device capable of generating output that may be intelligible to a user. The output devices **1210** may also include a touchscreen, a presence-sensitive display, or other input/output capable displays known in the art.

The network interface **1212** of the computer system **1200**, in some example embodiments, can be utilized to communicate with external devices via one or more data networks such as one or more wired, wireless, or optical networks including, for example, the Internet, intranet, LAN, WAN, cellular phone networks, Bluetooth radio, and an IEEE 902.11-based radio frequency network, Wi-Fi networks®, among others. The network interface **1212** may be a network interface card, such as an Ethernet card, an optical transceiver, a radio frequency transceiver, or any other type of device that can send and receive information.

The operating system **1214** may control one or more functionalities of the computer system **1200** and/or components thereof. For example, the operating system **1214** may interact with the software applications **1216** and may facilitate one or more interactions between the software applications **1216** and components of the computer system **1200**. As shown in FIG. **12**, the operating system **1214** may interact with or be otherwise coupled to the software applications **1216** and components thereof. In some embodiments, the software applications **1216** may be included in the operating system **1214**. In these and other examples, virtual modules, firmware, or software may be part of software applications **1216**.

Thus, systems and methods for real-time correction of accent in speech audio signals have been described. Although embodiments have been described with reference to specific example embodiments, it will be evident that various modifications and changes can be made to these example embodiments without departing from the broader spirit and scope of the present Application. Accordingly, the specification and drawings are to be regarded in an illustrative rather than a restrictive sense.

What is claimed is:

1. A method for real-time correction of an accent in a speech audio signal, the method being implemented by a computing device and comprising:

dividing the speech audio signal into a stream of input chunks, an input chunk from the stream of input chunks including a pre-defined number of frames of the speech audio signal;

extracting, by an acoustic features extraction module from the input chunk and a context associated with the input chunk, acoustic features;

extracting, by a linguistic features extraction module from the input chunk and the context, linguistic features with a reduced accent;

receiving a speaker embedding for a human speaker; and

generating an output chunk of an output audio signal based on the speaker embedding, the acoustic features, and the linguistic features.

2. The method for real-time correction of claim **1**, wherein:

the speech audio signal is recorded from a voice of a user by the computing device via an acoustic sensor; and

a delay between a first timestamp and a second timestamp is between 40 milliseconds and 300 milliseconds, wherein the first timestamp corresponds to a time when the chunk of the speech audio signal is recorded and the second timestamp corresponds to a time when the output chunk is generated.

3. The method for real-time correction of claim **1**, wherein the speaker embedding is generated based on the speech audio signal.

4. The method for real-time correction of claim **1**, wherein the speaker embedding is pretrained based on audio data including a recorded speech of a target speaker having a further accent.

5. The method for real-time correction of claim **1**, wherein the linguistic features include one of the following: phonetic posteriorgrams with a standardized phonetic dictionary and phonetic posteriorgrams with a data-driven phonetic library.

6. The method for real-time correction of claim **1**, wherein the linguistic features extraction module includes a neural network trained based on audio data to output the linguistic features with the reduced accent, the neural network being trained using a loss function to reduce, in the linguistic features, contributions due to a further accent present in the audio data.

7. The method for real-time correction of claim **1**, wherein the acoustic features include a pitch of the speech audio signal, an energy of the speech audio signal, and a value of a voice activity detector, the voice activity detector indicating absence of a human voice in the speech audio signal or presence of the human voice in the speech audio signal.

8. The method for real-time correction of claim 1, wherein the context is a pre-determined number of the frames belonging to chunks preceding the input chunk in the stream of input chunks.

9. The method for real-time correction of claim 1, wherein the generating the output chunk includes:

providing the speaker embedding, the acoustic features, and the linguistic features to a synthesis module to generate a melspectrogram with the reduced accent; and

providing the melspectrogram to a vocoder to generate the output chunk of the output audio signal.

10. The method for real-time correction of claim 9, wherein:

the synthesis module includes an encoder, a decoder, and a post-net module designed to improve an output of the decoder; and

generating the melspectrogram includes:

processing the linguistic features with the reduced accent by the encoder to generate hidden features;

combining the hidden features, the acoustic features, and the speaker embeddings to generate further features;

processing the further features by the decoder and the post-net module to generate the melspectrogram.

11. The method for real-time correction of claim 10, wherein:

the acoustic features are split into a stream of acoustic features chunks corresponding to the chunks in the stream of input chunks;

the linguistic features are split into a stream of linguistic features chunks corresponding to the chunks in the stream of input chunks;

the melspectrogram is split into a stream of melspectrogram chunks corresponding to the chunks in the stream of input chunks; and

a melspectrogram chunk of the stream of melspectrogram chunks is generated based on the following:

an acoustic features chunk of the stream of acoustic features chunks and acoustic features context including the pre-determined number of acoustic features frames belonging to acoustic features chunks preceding the acoustic features chunk in the stream of acoustic features chunks; and

a linguistic features chunk of the stream of melspectrogram chunks and linguistic features context including the pre-determined number of linguistic features frames belonging to linguistic features chunks preceding the linguistic features chunk in the stream of acoustic features chunks.

12. The method for real-time correction of claim 1, further comprising, prior to dividing the speech audio signal, processing the speech audio signal by a digital signal processing module to adjust one or more characteristics of the speech audio signal to improve extraction of the linguistic features and the acoustic features.

13. The method for real-time correction of claim 1, further comprising:

prior to dividing the speech audio signal, processing the speech audio signal by a digital signal processing module to adjust a loudness of the speech audio signal from a first level to a second level;

after generating the output chunk of the output audio signal, processing the output chunk by the digital signal processing module, to adjust the loudness of the output audio signal to the first level.

14. A computing apparatus comprising:

a processor; and

a memory storing instructions that, when executed by the processor, configure the apparatus to:

divide a speech audio signal including an accent into a stream of input chunks, an input chunk from the stream of input chunks including a pre-defined number of frames of the speech audio signal;

extract, by an acoustic features extraction module from the input chunk and a context associated with the input chunk, acoustic features;

extract, by a linguistic features extraction module from the input chunk and the context, linguistic features with a reduced accent;

receive a speaker embedding for a human speaker; and

generate an output chunk of an output audio signal based on the speaker embedding, the acoustic features, and the linguistic features.

15. The computing apparatus of claim 14, wherein:

the speech audio signal is recorded from a voice of a user by the computing apparatus via an acoustic sensor; and

a delay between a first timestamp and a second timestamp is between 40 milliseconds and 300 milliseconds, wherein the first timestamp corresponds to a time when the chunk of the speech audio signal is recorded and the second timestamp corresponds to a time when the output chunk is generated.

16. The computing apparatus of claim 14, wherein the speaker embedding is generated based on the speech audio signal.

17. The computing apparatus of claim 14, wherein the speaker embedding is pretrained based on audio data including a recorded speech of a target speaker having a further accent.

18. The computing apparatus of claim 14, wherein:

the linguistic features include one of the following: phonetic posteriorgrams with a standardized phonetic dictionary and phonetic posteriorgrams with a data-driven phonetic library; and

the acoustic features include a pitch of the speech audio signal, an energy of the speech audio signal, and a value of a voice activity detector, the voice activity detector indicating absence of a human voice in the speech audio signal or presence of the human voice in the speech audio signal.

19. The computing apparatus of claim 14, wherein the linguistic features extraction module includes a neural network trained based on audio data to output the linguistic features with the reduced accent, the neural network being trained using a loss function to reduce, in the linguistic features, contributions due to a further accent present in the audio data.

20. A non-transitory computer-readable storage medium, the computer-readable storage medium including instructions that when executed by a computer, cause the computer to:

divide a speech audio signal including an accent into a stream of input chunks, an input chunk from the stream of input chunks including a pre-defined number of frames of the speech audio signal;

extract, by an acoustic features extraction module from the input chunk and a context associated with the input chunk, acoustic features;

extract, by a linguistic features extraction module from the input chunk and the context, linguistic features with a reduced accent;

receive a speaker embedding for a human speaker; and

generate an output chunk of an output audio signal based on the speaker embedding, the acoustic features, and the linguistic features.

* * * * *